UNIVERSITY OF CAPE TOWN

MASTER THESIS

Automating User Privacy Policy Recommendations in Social Media

Author:
Ammar ABUELGASIM

Supervisor:
Dr. Anne KAYEM

A thesis submitted in fulfilment of the requirements for the degree of Master of Science

in the

Information Security Group
Department of Computer Science

Tuesday 31st May, 2016

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF.

Declaration of Authorship

I, Ammar ABUELGASIM, declare that this thesis titled, "Automating User Privacy Policy Recommendations in Social Media" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"The brick walls are there for a reason. The brick walls are not there to keep us out. The brick walls are there to give us a chance to show how badly we want something. Because the brick walls are there to stop the people who don't want it badly enough. They're there to stop the other people."

Randy Pausch

Acknowledgements

I would like to extend my sincere gratitude to my supervisor: Dr. Anne KAYEM, for the priceless support and guidance that she generously provided me through the whole of this project. Without her support and guidance, this project would not have been completed. Thank you Dr. Anne KAYEM for being the best supervisor a student can hope for.

My sincere appreciation is also extended to my family: Nimaat, Mohammed, and Osama for their unlimited motivation and support that kept me going through the “ups and downs" of this journey. Special thanks also go to my cousin Ibrahim, for being my home, thousands of miles away from home. Thank you all for being there.

I would also like to thank my colleagues in the Information Security Group and my fellow labmates, for the rich discussions, and the valuable feedback that made my journey far less painful than it could have been. Special thanks also go to my friend and colleague James Mutuku for the priceless advices, and the countless adventures.

Thank you all.

Ammar ABUELGASIM

List of Publications

1. AMMAR ABUELGASIM, AND ANNE V.D.M. KAYEM (2016), "An Approach to Personalized Privacy Recommendations on Online Social Networks", *In Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP 2016)*, Rome, Italy - Feb. 19-21, 2016. ISBN 978-989-758-167-0, pages 126-137. DOI: 10.5220/0005689701260137

2. AMMAR ABUELGASIM AND ANNE V.D.M. KAYEM (2016), "A Snow-Ball Algorithm for Automating Privacy Policy Configuration in Social Media", *Book: "Information Systems Security and Privacy", in Springer Lecture Series on Communications in Computer and Information Science*, (Under Review).

Contents

List of Figures

Dedicated to my loving family, Mohammed Abuelgasim, Nimaat Hamza, and Osama Abuelgasim. You taught me that I can.

Chapter 1

Introduction

1.1 Context and Motivation

Social media has revolutionised the way that users interact and communicate online by introducing new and innovative ways for self-expression, relationship formation, job seeking, and so on. Thanks to these advantages, social media platforms (SMP) attract massive numbers of users. For example, since its creation in 2004, Facebook has attracted 1.49 billion monthly active users. 968 million of who access Facebook daily [24]. Likewise, Twitter has drawn about 316 million users since 2006 [63].

These users disclose large volumes of information during social media interactions. For instance, every day about 350 million photos are shared on Facebook [27], and 500 million tweets sent on Twitter [63]. Most of this information is personal and sensitive in nature, as pointed out by Gross and Aquisti; and Stuttmann, *et al.* [32, 60].

The ease with which such personal information can be accessed, and potentially by a large number of people, exposes social media users to many privacy and security risks. Examples of such privacy and security risks include identity

theft, financial fraud, cyberstalking, cyberbullying, insurance and employment discrimination, embarrassment and losing face with friends [27, 28, 1].

In order to protect users' security and privacy, most SMPs have introduced privacy policies. Privacy policies are basically a set of rules to enable users to control who can access the information they disclose. These privacy policies can be fine-grained, such that SMP users can control access to individual profile attributes, for example address, birthdate, or cellphone number, as well as individual pieces of user-generated content, such as posts, status updates, photos, videos, and others. This is the case with SMPs like Google+ and Facebook.

If configured correctly, these privacy policies can enable social media users to regulate access to their personal and sensitive information, and thus to protect themselves against privacy violations. However, many existing studies have shown that numerous users fail to configure their privacy policies, either because they are not aware of the existence of these policies, or because the users find the privacy policies complex and time consuming [30, 44, 25, 47, 55].

1.2 Problem Statement

Various automated approaches have been proposed to assist users with privacy policy configuration [25, 55, 30, 4, 56, 52]. These approaches are, however, limited to either configuring privacy policies for profile attributes, or configuring privacy policies for user-generated content. This is problematic because both profile attributes and user-generated content can contain sensitive information. Therefore, protecting one without the other can still result in privacy violations. Furthermore, most of the proposed privacy policy configuration approaches require considerable user input, which is a time-consuming process.

The research revealed no existing solution for automating privacy policies that caters for both profile attributes and user-generated content. Therefore, the goal of this thesis is to propose an approach to privacy policy automation that: (1) handles both profile attributes and user generated content; (2) requires only minimum user input. Such an approach would provide better privacy protection than existing privacy policy automation solutions.

1.3 Proposed Solution

In order to alleviate the deficit that many social media users face with respect to privacy policy configuration, this study proposes a privacy policy recommender system. This recommender system utilises minimum input from social media users and, in return, it provides these social media users with personalised suggestions about how they (i.e. users) should configure the privacy policies of their profile attributes as well as their generated content.

The privacy policy recommender system consists of two independent components. The first component, termed the profile attributes protector (PAP), is responsible for suggesting suitable privacy policies for the users' profile attributes. The PAP utilises privacy policies that existing (presumably more experienced) social media users have configured for their profile attributes to suggest to novice (presumably naïve) social media users as how to configure the privacy policies for their profile attributes. The second component of the recommender, termed the user content protector (UCP), is responsible for suggesting suitable privacy policies for the users' generated content. The PAP 'learns' from users' privacy policy history (i.e. the privacy policies that the particular user has configured for his/her previously generated content), and uses this 'knowledge' to suggest privacy policies for the content that users might share in the future.

The advantage of this solution is twofold. First, both profile attributes and user-generated content is protected, which is important in preventing privacy leaks. Second, by providing the users with personalised privacy policy suggestions, the issue of manual privacy policy configuration is alleviated.

1.4 Thesis Outline

The rest of this thesis is structured as follows. In Chapter 2, the background work related to the area of social media and privacy policies is presented and discussed. Chapter 3 follows this with a specification of a formal model for describing user relationships on SMPs and the concepts that underpin privacy policy configuration. Chapter 4 builds on the formal specifications given in Chapter 3 to describe the theoretical framework of the privacy policy recommender system. Chapter 5 presents experimental results from prototype implementation of the privacy policy recommender system. Finally, in Chapter 6 the study is concluded with a summary of the contributions by the research and ideas are offered for future work.

Chapter 2

Background

2.1 Introduction

In this chapter, the relevant background work related to the area of social media and privacy policies is presented and discussed. In addition, it presents concepts to aid understanding of the proposed privacy policy recommender system, presented in Chapter 4.

2.2 Social Media Platforms

The concept of social networks – "a set of people or other social entities connected by a set of socially meaningful relationships" [68] – is not a recent one. Historically, humans have been known to create intricate networks of diverse relationships between individuals, families and tribes.

However, the advances in information and communication technology and the emergence of the World Wide Web (www) have allowed social networks to transcend from the physical world to the realm of online. This has given rise to various forms of electronic social media platforms (SMPs). These SMPs can be broadly defined as:

> *"Any web-based (or electronic) services that allow people to: create profile that describes them (i.e. the people); formulate and maintain a social relationships with other people within the same service; and view and traverse the profiles and social relationships made by others within the same service."* [13, 59, 35, 64]

The history of social media dates back to 1997 and the launch of the first 'recognisable' SMP of SixDegrees.com. SixDegrees.com was the first website to combine all the features of a SMP, such as creating profiles, articulating friendships, and traversing friends lists [13]. Since the introduction of SixDegrees.com, the market has been flooded with new SMPs, witnessing many short-lived success stories (e.g. Friendster and MySpace) and many stories of failure (e.g. Orkut and Windows Live Spaces). Most of these early SMPs did not survive due to technical, financial or social problems.

Nonetheless, these early SMPs paved the way for the next generation of SMPs, which have circumvented the pitfalls of the previous generation and grown into a global phenomenon with an unmistakable social and economic impact [35], as attested by SMPs like Facebook and Google+, which serve millions of users globally and generate billions of dollars in revenue [35, 63, 24].

Nowadays, SMPs occupy a considerable portion of people's online activities. People take part in different types of SMPs ranging from general purpose SMPs that provide their services to the greater public (e.g. Facebook and Google+), to more exclusive SMPs that are built around a specific focus (e.g. LinkedIn, Ryze, XING and ResearchGate).

The next section discusses the notion of privacy in general and roughly defines privacy in the context of social media.

2.3 The Concept of Privacy

The notion of privacy has evolved over the years, from early monotonic conceptualisations of privacy to recent views of privacy as a balancing act between disclosure and concealment [38]. One such recent view is provided by Altman, who defines privacy as "*the selective control of access to the self*" [5]. Altman views privacy as a combination of three processes: a *dialectic* process that involves both disclosing and withholding information, an *optimisation* process that seeks an optimum level of disclosure, and a *multi-modal* process that utilises different verbal, behavioural and environmental mechanisms.

Privacy is important in human societies: it is necessary for regulating social interactions between people and evolving individuals' self-identity [5]. This is why privacy as a generic process has been observed in all human societies, but its enforcement mechanisms may differ from one society to another [5].

SMPs are human societies and are thus not an exception when it comes to the need for privacy. In the context of SMPs, privacy can be defined as a user's ability to selectively control and regulate access to the information he/she personally generates, or to any information related to him/her within an SMP. This access can be by other users within the scope of the SMP, or by some other external entity. Every time the control over this content is jeopardised, that user's privacy is violated.

In the subsequent section, the literature on privacy in social media is surveyed, highlighting the main research focuses within the area.

2.4 Social Media and Privacy

The emergence of SMPs and the explosion of personal information that followed have opened a Pandora's box of privacy and security issues, revealing a plethora of ways in which users' privacy and security can be jeopardised.

Consequently, a large body of research has been dedicated to addressing these privacy and security issues. These studies can be roughly grouped into six categories, namely publishing social graph data, malicious third-party applications, privacy attacks in SMPs, measuring privacy in SMPs, access control models for SMPs, and privacy policies' automation.

The following subsections briefly discuss and present examples of studies that fall within each of the above-mentioned categories.

2.4.1 Publishing Social Graph Data

The collection of users' profiles and the set of relationships between them in an SMP are usually modelled as a graph, where nodes correspond to users' profiles, and edges to a social relationship between two user's profiles. Such a graph is referred to as a social graph [41].

Social graph data is of interest to many third parties (i.e. external entities), since it can be mined to reveal insightful information to businesses [31], advertisers [22], as well as researchers. Therefore, SMP providers may wish to publish social graph data to these third parties. However, simply releasing such sensitive data to presumably untrusted external entities jeopardises users' security and privacy. As a result, many researchers have investigated the issue of publishing social graph data in a way that maintains the anonymity of the data and hence the privacy of users.

For instance, Campan and Truta [14] argue that the traditional way of anonymising the social graph, by simply removing identifying information, is not enough. Campan and Truta explain that adversaries can still re-identify users and gain access to users' sensitive information. For example, using background knowledge about the neighbourhood structure of users in the social graph or by combining background knowledge about the semi-identifiable attributes (i.e. quasi-identifiers) of users [7].

In order to resolve this issue, Campan and Truta [14] reintroduced the notion of k-anonymity to the social media context by dividing the graph into a set of disjoint clusters of the size of at least k, where nodes within the cluster are generalised so that they are indistinguishable from each other. Each cluster is then collapsed into one 'generalised' node. Next, a link is formed between two clusters (i.e. generalised nodes) if they (i.e. the clusters) are not separate.

Similarly Wei and Lu [67] show that even if social graph data is anonymised by removing identifying information and generalising node labels, adversarial re-identification of users in the anonymised graph is still possible with some background knowledge about the victim's neighbourhood structure. They proposed *k-subgraphs* as a solution that combines label anonymisation with structural anonymisation to limit the risk of privacy disclosure in social media data publication. The *k-subgraph* operates in three phases: first, the social graph is partitioned into a set of disjoint subgraphs, such that each node within the subgraph has the same label. Second, the degrees of each node in the same subgraph are unified. Third, disjointed subgraphs are connected if the nodes embedded therein are connected.

On the other hand, Yuan, *et al.* [71] argue that anonymising the social graph against one level of background knowledge attacks does not meet personalised privacy requirements. In fact, it can decrease the utility of the social graph (i.e.

how much it resembles the original graph). Yuan, *et al.* propose a multi-level anonymisation scheme to prevent nodes re-identification, while at the same time maintaining the utility of the social graph.

Social graph anonymisation is an important privacy issue. However, it is only concerned with the privacy of users when their data is being shared with third parties. Little attention is given to the privacy of users in their daily interactions within SMPs, which is when many of the privacy risks emerge.

2.4.2 Malicious Third Party Applications

In August 2006, Facebook released the first version of their application programming interface (API) [24], soon followed by Google+ and others. These APIs enable third-party developers to build applications that provide SMP users with exciting new functionalities ranging from gaming (e.g. Farmville and Candy Crush) to sending gifts to friends (e.g. birthday cards).

However, these applications raised several privacy issues because they have access to users' information and are not bound by the privacy agreements of the SMP provider. This is a serious problem because these applications can aggregate user data and store this data on external servers, where it can be manipulated at the discretion of third parties.

Felt and Evans [26] studied 150 popular Facebook applications and found that most have more access than they actually need to function. Consequently, Felt and Evans proposed a *privacy-by-proxy* solution, whereby third-party applications do not directly access users' information. Instead they interact with the data through special markup tags.

Besmer, *et al.* [10] criticise the *privacy-by-proxy* approach, arguing that it "severely limits the social value of many applications" [10] and, even if it did not, it

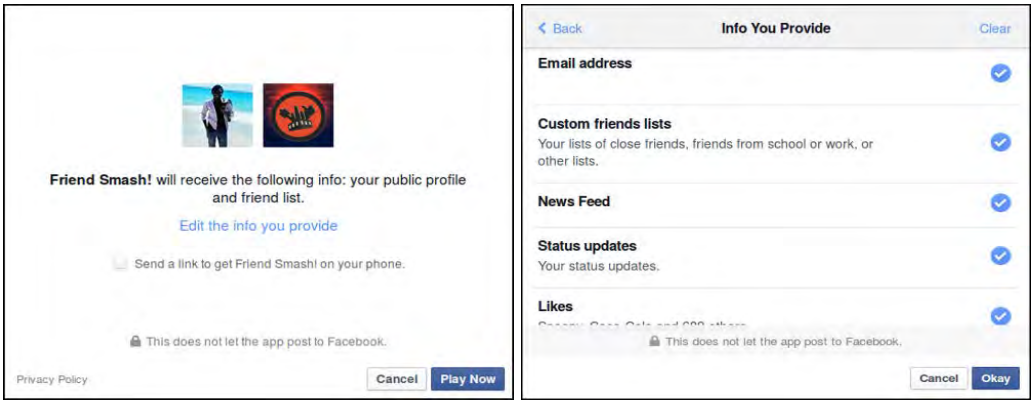
"forces the developer to expose their business logic, usually in the form of JavaScript, to the social network and its users" [10]. Furthermore, they argue that current SMP architectures do not allow users to specify which information these applications can access. Besmer, *et al.* suggest the alternative of adding a new *user-to-application policy*, whereby users can decide which information the application can access. This new policy, combined with the existing default application policy, preserves users' privacy while minimally changing the existing architecture.

Wang, *et al.* [65] inspected 1,800 Facebook applications and noticed that upon installation users cannot deny permission requests made by an application. In addition, these permission requests do not reflect the actual scope of information the application can access. Furthermore, they noticed that the applications could override users' privacy policies. In order to alleviate these issues, Wang, *et al.* propose a new application authentication dialogue that enables users to limit applications' information access and notify users when applications' behaviour might violate users' global privacy policies.

On the other hand, Cheng, *et al.* [20] followed an entirely different approach by partitioning third-party applications into *external components*, which are controlled by the third party, and *internal components*, which are controlled by the SMP. The modules running on the internal component can access users' sensitive information but cannot transmit this information outside the SMP. This not only ensures that applications get the required access to users information, but also that the users' privacy remains protected.

It is worth mentioning that the release of the Facebook API v.2 addresses many of the issues raised above. For example, users now can specify fine-grained access permissions for each application they install in their profiles, as shown in Figure 2.1. The new API also severely limits applications' abilities to collect

information about users and their friends.



(a) *Application authentication dialog* ***(b)*** *Requested permissions by the app.*

Figure 2.1: *Facebook's updates regarding applications' access permissions*

2.4.3 Privacy Attacks on Social Media Platforms

A sizeable body of knowledge is dedicated to various attacks that expose vulnerabilities in the standard SMPs' privacy schemes, and pointing out solutions to mitigating these potential risks. Such privacy attacks can be more or less categorised into three categories: graph approximation, attribute inference, and identity clone attacks.

2.4.3.1 Graph Approximation Attacks

In graph approximation attacks, the adversary tries to reconstruct the entire social graph using locally available information for the purpose of gaining valuable graph statistics, such as node centrality or degree distribution, amongst others. The work of Bonneau, *et al.* [12] exemplifies this category. Bonneau, *et al.* reconstructed the social graph with high accuracy, using available public listing information.

Likewise, Jin, *et al.* [37] developed an attack whereby an adversary uses friend lists that are visible to him/her to compromise the friend list of a targeted victim

user, and possibly reconstruct the entire social graph. Figure 2.2 below shows an oversimplified version of this attack. In this scenario, *Bob* is the adversary and *Alice* is the target. *Alice* has configured her privacy policies so that only her direct friends can see her friend list. But if *Bob* manages to befriend *Frank* and *Eve*, he can gain access to their friend lists, and thus infer that they both are friends of *Alice*, which is exactly the information that *Alice* was trying to hide.

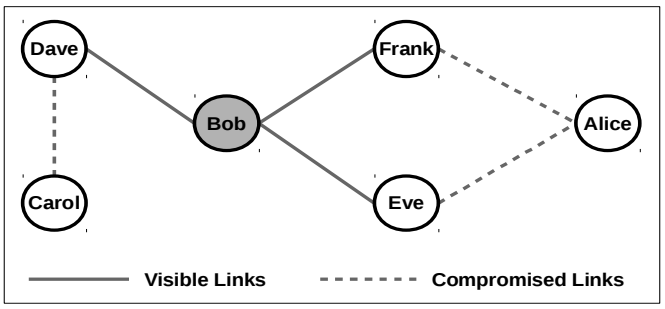


Figure 2.2: *A simple example of a link-inference attack.*

Similarly, Yap, *et al.* [23] show that by bribing a small set of nodes in the SMPs' graph, attackers could compromise the friendship structure of a much larger set of nodes, giving them a wider view of the entire graph. According to Yap, *et al.* [23], such attacks can be prevented or at least severely limited by hiding friend lists and 'degree information', both of which can be achieved by having more restrictive privacy policies for friend lists.

2.4.3.2 Attribute Inference Attacks

Another category of attacks is attribute inference attacks, whereby adversaries try to predict the value of the victim's private (i.e. hidden) profile attributes. Within this category, Zheleva and Getoor [72] show that hidden attributes of private profile users can be inferred using the attributes of their friends who have made their profile information publicly available. For example, if *Bob* hid his nationality, but at the same time the majority of his friends were South Africans, adversaries can infer that *Bob* is probably also a South African.

In the same direction, Pesce, *et al.* [50] argue that people who are tagged in a photo have probably shared the same place for some time, and that taking a photograph with a person is more socially relevant than just knowing him/her. Accordingly, Pesce, *et al.* have shown that attribute inference attacks can be improved considerably by incorporating photo-tagging information.

Yang, *et al.* [70] follow a slightly different approach. They carried out an attribute-based re-identification attack, whereby an attacker knows only a small 'seed' of the target user's public attributes, and uses this 'seed' to re-identify the target user from within a large database of users' information, thus gaining access to the rest of the target's attributes. Yang, *et al.* defines two types of attackers, namely *resourceful attackers*, who have the skills and resources to build a local database of user-related information (typically by crawling SMPs and public records), and to re-identify targets (i.e. victims) by checking their seeds against this database. The second type of attackers are *tireless attackers* who compensate for their lack of ability to build large databases of user-related information by investing more time and effort into sending the 'seed' information to search engines, and then tirelessly sift through the results.

2.4.3.3 Identity Clone Attacks

Identity clone attacks are a type of attack whereby adversaries make a copy (i.e. clone) of the victim's profile in the SMP in order to befriend his/her friends and gain access to their information. An example of this category is in the work of Bilge, *et al.* [11], who successfully provoked an automated profile cloning attack. First, they crawled a number of publicly available profiles and then used data from the crawled profiles to automatically make clones. In turn, the clones sent automatic friendship requests to victims. Furthermore, Bilge, *et al.*

have shown that the original and cloned profile do not have to be on the same SMP, and that cross-site identity cloning is possible.

Shan, *et al.* [54] improve upon the original identity cloning attack by adopting two novel strategies, namely snowball sampling and iteration, to increase the request acceptance rate and create more credible clones. To solve this issue, Shan, *et al.* proposes *CloneSpotter*, a server side solution that relies on IP addresses to detect clones.

2.4.4 Measuring Privacy in Social Media Platforms

Several studies have focused on developing various measures for privacy, information leakage, and risk in SMPs, with the sole purpose of enabling SMP users to make better-informed privacy decisions. Baker and Chen [9] argue that measuring information leakage in SMPs is necessary, particularly due to the scale of information being shared. Therefore, they propose *PrivAware* as a solution that measures information leakage in a user's profile, based on the profile's number of 'inferable' attributes. Furthermore, *PrivAware* suggests several actions that users can take to mitigate leakage, such as removing risky friends.

Talukder, *et al.* [61] argue that even if users hide their sensitive information it can still be inferred (i.e. leaked) using publicly available information from the users' friends, group membership, photo tags, etc. To solve the problem, Talukder, *et al.* propose *Privometer*, which is an information leakage measurement tool. *Privometer* tries to proactively infer the user's sensitive attributes and ranks the user's friends according to their contribution to the sensitive information leak. *Privometer* also suggests 'self-sanitisation' actions to the user, which include removing those friends with the highest contribution to sensitive information leakage.

Liu and Terzi [43] argue that most of the research on privacy in SMPs is focused on 'corporate-scale' privacy concerns and that little research is channelled towards addressing privacy risks that stem from users' information-sharing activities in SMPs. Therefore, Liu and Terzi introduce a framework for calculating a *privacy score* for SMP users. This privacy score measures the user's privacy risk stemming from his/her information sharing behaviour, whereby the more sensitive the information and the more people who see that information, the higher the privacy score. The privacy score of a user is the sum of the privacy score of his/her profile items, and the privacy score of a profile item is a function of that item's *sensitivity* and *visibility*.

Srivastava and Geethakumari [58] argue that there is no current measure that enables users to evaluate their privacy situation. Therefore, Srivastava and Geethakumari propose calculating the *privacy quotient*, which is a real value that quantifies the user's attitude towards privacy. The privacy quotient of a user is the sum of the privacy quotients of his/her profile attributes and the privacy quotients of a single profile attribute is a combination of the attribute's sensitivity, and its visibility in the SMP.

Wang, *et al.* [66] postulate that, without a practical way to quantify and measure privacy, it will be difficult for users to decide how much risk they are willing to take, or to come up with appropriate policies to protect their privacy. Therefore, Wang, *et al.* propose *Privacy Index* (PIDX), which is a measure that quantifies users' privacy exposure to users based on their visible profile attributes to other users. Where, PIDX $\in [0, 100]$, such that high PIDX value indicates high privacy exposure and low PIDX values indicate low privacy exposure.

Akcora, *et al.* [3] argue that there is no measure that informs users how risky it is to form a friendship with a stranger in SMPs, even though such friendships can have serious consequences for user privacy. Therefore, Akcora, *et al.* propose

a framework for assessing the 'riskiness' of strangers (i.e. potential friends) to help users in judging strangers before befriending them. In order to assess the riskiness of a stranger, this framework considers social graph properties, profile similarities, and the risks/benefits of befriending that stranger.

2.4.5 Access Control for Social Media Platforms

The access control model adopted by an SMP has a strong effect on users' privacy as it dictates how they can regulate access to their information and to what extent. Therefore, many researchers have worked on better ways to enforce access control in SMPs.

Traditionally, most SMPs adopted an access control model that is similar to discretionary access control, whereby content owners can specify policies to regulate access to their content, and access rights are granted based on the existence of a relationship between the accessor and the content owner.

Cheng, *et al.* [19], however, argue that this model neglects the fact that in SMPs multiple types of relationships co-exist in the same ego-network and, by not supporting multiple relationship types, the current model severely limits the expressiveness of its access control policies. Consequently, Cheng, *et al.* [19] propose a User-to-User Relationship-Based Access Control (UURAC) model for SMPs. This model supports multiple types of relationships, users and resources as targets, as well as user policies for outgoing and incoming actions utilising regular expression notation. Furthermore, Cheng, *et al.* [18] refine the UURAC model by adding user-to-resource and resource-to-resource relationships. These provide even more access expressive control policies.

On the other hand, Carminati, *et al.* [17] contend that SMP providers are not to be trusted with the enforcement of access control, citing several incidents where

SMP providers violated the privacy of their own users (e.g. Facebook Beacon application). Therefore, Carminati, *et al.* propose a semi-decentralised architecture for SMPs, enforcing access control on the client side. This access control model is based on trust levels, relationship types and relationship depths.

In another article, Carminati, *et al.* [15, 16] argue that improving the SMPs' access control is the first step toward addressing SMP security and privacy issues. Carminati, *et al.* criticise the existing access control models for being very basic and lacking flexibility. Rather, they propose a semantic web-based access control model that provides users with different types of policies, namely *access control polices* that enable content's owners and participants to specify who can access this content; *filtering policies* that enable users to specify which type of content should be filtered out when browsing the SMP; and *admin policies* that enable the SMP's administrator to determine things such as who can specify access control policies and for which content.

Similar to the approach of Carminati, *et al.* [15, 16], Masoumzadeh and Joshi [48] argue that SMPs contain intricate semantic relationships among users, data objects, and between users and data objects, and that most of the proposed access control models – including the ontology-based models – do not take these complexities into consideration. Therefore, Masoumzadeh and Joshi propose an Ontology-Based Social Network Access Control (OSNAC) that supports high-level system-wide policies, as well as advanced user-level policies that enable users to flexibly control their information.

Many of the access control models proposed above aim to provide more expressive access control policies (i.e. privacy policies) to enable users to better control their information exposure. Through incorporating additional elements, like trust level and relationship type, or using complex ontologies, these new models provide flexible, expressive policies. However, as these models become

more sophisticated, the corresponding privacy policies become more complicated, adding to the existing overhead placed on end users.

2.4.6 Privacy Policies Automation

In harmony with this current research, many researchers focus on privacy policies in SMPs due in part to their importance as tools for privacy protection, but mostly for the infamous reputation these policies have for being complex and time consuming [30, 44, 25, 47, 55]. Subsequently, many researchers have proposed some form of automation to eliminate or at least alleviate the overhead associated with manually configuring privacy policies.

Guo and Chen [33] use item response theory (IRT) to explore the probabilistic relationship between users' privacy policies, and their level of utility and privacy concern. Guo and Chen then recommend privacy policies that satisfy an optimum level of utility for a given privacy concern level. Guo and Chen's approach seems to automate the privacy policy configuration process to a high degree, as the only input required is the level of privacy concern. However, privacy concern level is not measurable, and is subject to the judgement of individuals. Furthermore, their approach does not recommend privacy policies for user-generated content.

Fang and LeFevre [25] propose a mechanism for configuring privacy policies. The privacy policy configuration mechanism follows an active learning approach, through which the user is prompted to label a subset of his/her friends for each profile attribute by stating whether or not the friends are allowed to access that profile attribute or not. Next, the mechanism uses this subset of friends to train a classifier that predicts which of the user's remaining friends are allowed (or not allowed) to access that particular profile attribute. While this approach facilitates setting fine-grained privacy policies, users are required

to provide considerable input to enable the system to run efficiently. In fact, for every profile attribute (and one can have up to 27 attributes) the user is required to manually label a group of friends in order to train the attribute's classifier. A further caveat of this solution is that it does not handle privacy policies for user-generated content, since the classifiers are trained to predict privacy policies for profile attributes only.

Similar to the solution of Fang and LeFevre [25], Shehab, *et al.* [55] propose a solution by which the user is required to label a selected subset of his/her friends as trusted or not trusted to access a particular profile object. This subset is then used to train a classifier that predicts which of the remaining friends are trusted (or not trusted) to access that particular object. In contrast to Fang and LeFevre's [25] scheme, the Shehab, *et al.* scheme introduces a new concept, merging the resulting classifier with other neighbouring users' classifiers to enhance the classifier's performance. However, the Shehab, *et al.* scheme also requires substantial user input because the user has to manually label a group of his/her friends for every profile object.

Toch, *et al.* [62] introduce the *Collaborative Policy Analysis* algorithm that analyses existing privacy policies to recommend personalised default privacy policies for new users in a location-sharing SMP. *Collaborative Policy Analysis* works as follows: first, it measures the similarity between every pair of existing policies by relying on the overlap in the geographical area covered by each policy, second, it uses the K-means algorithm to divide existing policies into distinct clusters. Lastly, it compares the new user's information with the information of the users in each policy cluster, and then recommends policies from the most relevant cluster. However, calculating policies' similarity might not always be as convenient as it is in the case of location-sharing SMPs. Furthermore, Toch, *et al.* do not provide any evaluation mechanism for the proposed solution.

Danezis [21] proposes a solution that exploits the concept of 'contextual integrity'. In this solution, privacy is maintained when the information is bound within a specific context (i.e. group of users). Danezis uses social network analysis (SNA) techniques to extract all of the possible contexts from the user's friendship network. Next, every piece of generated content is automatically assigned to a possible context (or contexts). Privacy is ensured by a default system-wide policy stating that only users in the context of an action or content are allowed to access it. This solution offers privacy with minimum input from the user, but at the same time one of its main pillars, which is content-to-context assignment, is still not well defined, especially regarding content types that cannot be easily linked with a context.

Ghazinour, *et al.* [29, 30] introduces YourPrivacyProtector, which is a solution that uses the K-Nearest Neighbours algorithm to find the closest three profiles to the user (i.e. neighbours), and then uses the privacy policies of the neighbouring users to suggest to the user whether or not to disclose a particular profile attribute. Nonetheless, this approach provides coarse-grained privacy policy suggestions, as it only advises users to disclose or hide attributes. Furthermore, this approach cannot support privacy policies for user-generated content because, unlike profile attributes, users might generate different types of content. For instance, Bob's status updates are usually very different from $Alice$'s status updates. Thus, it is not feasible to rely on other users' content policies for providing suggestions.

Alsalibi and Zakaria [4] propose a collaborative filtering privacy recommender system. In order to recommend privacy policies to a particular target user, this system first identifies a group of similar users (to the target), and then uses the most frequently used privacy policies within this group to make privacy policy

recommendations to the target user. However, this system cannot handle privacy policies for user-generated content because, like the approach of Ghazinour, *et al.*, it is not feasible to rely on other users' content policies for providing suggestions.

On the other hand, Jones [38] argues that current privacy mechanisms provided by SMPs fail to capture the complexity of users' relationships in terms of type and strength, resulting in a huge boundary regulation problem within users' networks and causing many users to over-share information or, in the other extreme, not share any information at all. Jones claims that dividing users' friends into homogeneous groups that share common interest or ties helps users to manage disclosure within their own networks. Consequently, Jones proposes an algorithm for automatically extracting and labelling such groups, and further studies how those groups can be automatically used to control personal information disclosure.

Sinha, *et al.* [56] argue that the increasing amount of content users share in SMPs increases the chance of users sharing content with an unintended audience. Thus, Sinha, *et al.* propose an automated tool to help users configure privacy policies for the text-based content they generate. They use supervised machine learning, particularly the MaxEnt (Maximum Entropy) classification algorithm to predict (and then recommend) privacy policies for text-based content. However, despite being able to predict privacy policies for textual content, this solution does not address the issue of configuring privacy policies for profile attributes.

Along the same lines, Sánchez and Viejo [52] propose an automated mechanism to inform SMP users about the privacy risks inherent to their unstructured text-based content, in order to enable users to make more informed privacy policy choices. The proposed mechanism adopts an information theoretic approach,

and works by comparing the text-based content's 'sensitivity' against the content owner's privacy requirements for all types of users in the SMP. However, this approach only warns users about potential privacy conflicts within their generated content and does not suggest privacy policies directly to them.

The next section takes a closer look at the important topic of recommender systems, which is closely related to the privacy policy recommender system proposed in this study.

2.5 Recommender Systems

Recommender Systems (RSs) or Recommendation Systems are a "class of web applications that involve predicting user responses to options" [42]. RSs are also defined as "software tools and techniques providing suggestions for items to be of use to a user" [51].

Traditionally, RSs have been used in e-commerce as they enable online shop owners to increase their sales and market more diverse products that are otherwise impossible to market in traditional brick-and-mortar shops [42]. For example, sites like Amazon attribute a large percentage of their sales to RSs. However, the uses of RSs are not limited to e-commerce only, as they are also used in other fields, such as in recommending news articles (e.g. personalised newspapers), travel destinations (e.g. Tripadvisor), movies (e.g. Netflix, YouTube), and courses and books (e.g. Coursera) [51, 42].

2.5.1 The General Setup for Recommender Systems

The general setup for RSs (i.e. the environment within which RSs are formulated) consists of three classes of entities, namely *users* who are the recipients of recommendations; *items*, which are the entities being recommended to users;

and, lastly, user-item *ratings* that represent the utility (i.e. value) of a specific item to a user. Users may have a set of features that describe them (e.g. demographics), and items also can have a set of features describing them (e.g. colour or genre). Figure 2.3 below depicts the general setup for RSs.

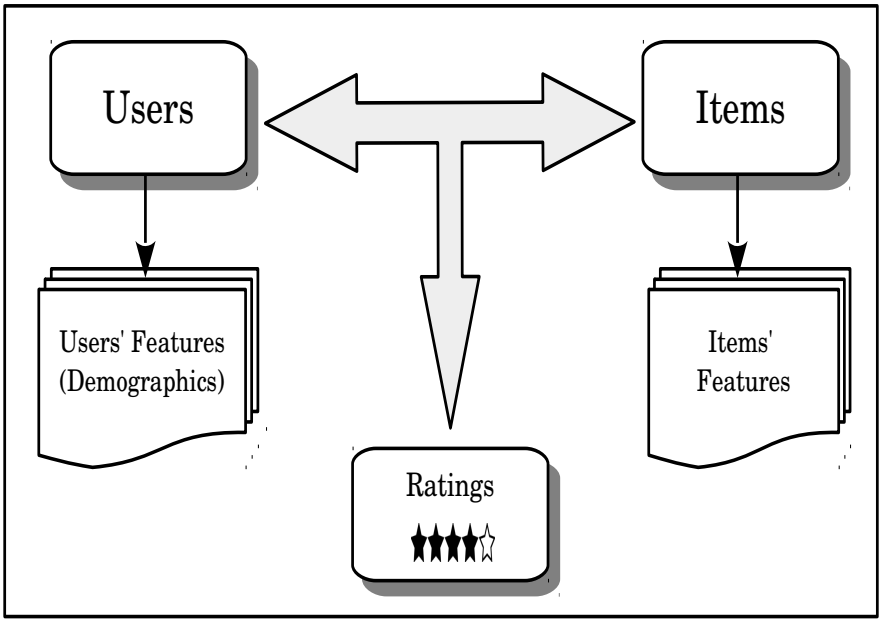


Figure 2.3: *The General Setup for Recommender Systems (source: Konstan [40]).*

2.5.2 How Recommender Systems Work

In general, all RSs follow three steps in order to provide recommendations for their users, who are usually referred to as *target users*. The first step is collecting some form of background data about users, items and, of course, user-item ratings. This background data might vary according to the methodology used for building the RS [51]. An example of background data – in the case of a book RS – might be readers' characteristics (e.g. age and gender), books' characteristics (e.g. title and author), in addition to the reader-book ratings.

The second step is utilising this background data to predict how target users would rate some formerly unrated or unknown items [42]. The third and final step is suggesting the items that receive the highest predicted ratings to target

users. These suggestions can be presented as explicit recommendations to target users, or implicitly by, for instance, placing the suggested items strategically on the screen so that users can spot them easily.

Almost all RSs implement the first and the third step in the same way. However, in order to implement the second step, which is predicting users' ratings for items, RSs follow several methodologies, for instance collaborative filtering, content-based filtering, demographic-based filtering, and so on. These methodologies are briefly explained below.

2.5.2.1 Collaborative Filtering (CF)

Collaborative Filtering RSs rely on the assumption that past agreements predict future agreements [40]. Thus, in order to predict the target user's rating for a specific item, CF-RSs look at how other users with similar tastes (i.e. ratings history) have rated that item.

For instance, as shown in Figure 2.4 below, in order to predict whether Bob will like the book *The Da Vinci Code* or not, the collaborative filtering RS looks at how other readers with tastes similar to Bob's (i.e. users who liked/disliked the same books that Bob liked/disliked) have rated *The Da Vinci Code* [42, 51].

Figure 2.4: *How Collaborative Filtering RSs provide recommendations to users.*

2.5.2.2 Content-Based Filtering (CB)

Content-Based RSs, on the other hand, assume that similar items receive similar ratings. So, in order to predict the active user's rating for an item, content-based RSs look at the items that the target user has rated previously and predict the current item's rating based on the similarity between it and those items.

For instance, as shown in Figure 2.5 below, if Bob previously liked *The Da Vinci Code* and *Angels & Demons*, the content-based RS will predict that Bob might also like *The Lost Symbol*[1] [42, 51].

Figure 2.5: *How Content-Based RSs provide recommendations to users.*

2.5.2.3 Demographic-Based (DB)

Demographic-Based RSs, on the other hand, rely on users' demographic data (e.g. age, gender, education, etc.) to predict users' ratings. Demographic-based RSs rely on the assumption that demographically similar users are more likely to rate items similarly.

[1] These books: *The Da Vinci Code*, *Angles & Demons*, and *The Lost Symbol* are similar, they were all written by *Dan Brown*, and are part of the *Robert Langdon* series

For instance, as shown in Figure 2.6 below, if many demographically similar readers to the target users have liked certain books, the demographic-based RS will recommend these books for target users [51, 36].

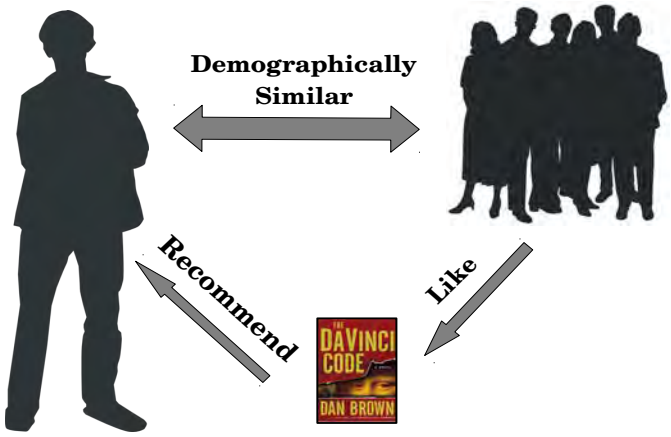


Figure 2.6: *How Demographic-Based RSs provide recommendations to users.*

2.5.2.4 Non-Personalised (NP)

Non-personalised RSs are not customised for a specific target user, and usually predict users' ratings by relying on the majority vote. For example, recommending the top ten best-selling books to readers [51, 36].

2.5.2.5 Hybrid Recommender Systems

As indicated by their names, hybrid recommender systems are a combination of the previously mentioned RSs methodologies [51].

In the following section, the concept of text classification is explained, which is a concept relevant to understanding this study's proposed privacy policy recommender system.

2.6 Text Classification

Text Classification (TC) is "the activity of labelling natural language text with thematic categories from a predefined set of categories" [53]. TC is also known as text categorisation and topic spotting [53].

Generally speaking, in TC there is a set of documents $\Omega = \{d_1, d_2, \ldots, d_m\}$ (usually referred to as *corpus*) where each document $d_i \in \Omega$ is assigned a category (i.e. class) from a predefined set of categories $C = \{c_1, c_2, \ldots, c_k\}$. For instance, documents can be emails, and the categories specify whether an email is spam or not. Or, in a different scenario, documents can be news articles, and categories can be whether an article is an economic, political, or sports article.

This text corpus is then *preprocessed*, such that each document d_i is transformed into a set of features that describe the document. In TC, preprocessing methods usually consist of a combination of the following steps [2, 53]:

1. **Tokenisation:** During this step the text corpus Ω is broken down into small units called terms t. Terms can be individual words, as in the case of *word tokenisation*, or a sequence of words, as in the case of *n-gram tokenisation*. The set of all terms created by tokenising Ω is $\mathbb{V} = \{t_1, t_2, \ldots, t_{|\mathbb{V}|}\}$, and is commonly referred to as the vocabulary.

2. **Normalisation:** In this optional step, all terms are stripped of their diacritical marks (e.g. ä or â become a), lower-cased, and kept in their normal form. This is in order to avoid treating the same word as a different word.

3. **Stop Words Removal:** This is an optional step, within which stop words (like 'a', 'if' or 'in') are removed from the vocabulary $\mathbb{V}$. This is because they are considered neutral and do not contribute to the classification task.

4. **Stemming:** This is also an optional step, within which all words are reduced to their stem (i.e. basic form) by removing affixes like 'ed', 'ing', etc.

5. **Vectorisation:** In this step, each text document $d_i \in \Omega$ is represented as a $|\mathbb{V}|$-dimensional vector $d_i = \langle f_{i1}, f_{i2}, \ldots, f_{i|\mathbb{V}|} \rangle$. Where, each item f_{ij} can either be a binary value representing the existence of term t_j in d_i; or a weighted value that represents the importance of term t_j in d_i.

   Term Frequency-Inverse Document Frequency (tf-idf), is a widely used method for calculating terms' weights, and it is calculated by the following:

$$tf\text{-}idf(t_k, d_i) = tf(t_k, d_i) * \log\left(\frac{\text{No. all documents}}{df(t_k)}\right) \tag{2.1}$$

   Where $tf(t_k, d_i)$ = The number of times the term t appears in document d_i, While $df(t_k)$ = The number of documents that contain term t_k.

6. **Dimentionality Reduction:** In this step, the dimensionality of the resulting feature vector is reduced, by selecting an *informative* subset of the terms from $\mathbb{V}$ instead of using all the terms in $\mathbb{V}$ [53].

Figure 2.7 below, shows the general flow of the preprocessing process.

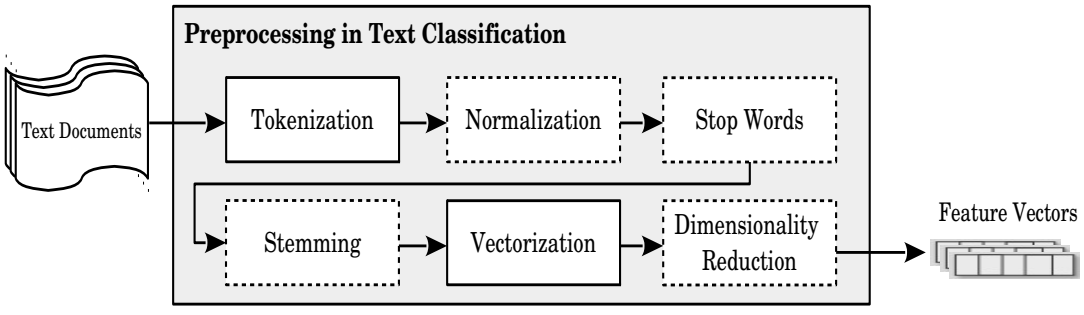


Figure 2.7: *The preprocessing phase, optional steps are in dotted boxes.*

After preprocessing the initial text corpus Ω, a machine-learning classification algorithm (e.g. Naïve Bayes, SVM, etc.), is applied to the preprocessed text

corpus (which is denoted as $\hat{\Omega}$) to inductively build a *classifier* that associates documents' underlying characteristics to categories. This *classifier* is then used to predict the categories of uncategorised documents [2], which it does by mapping the features of the uncategorised document to a particular category.

TC has a wide range of applications, including email and spam filtering; opinion mining and sentiment analysis; document organisation; news filtering and organisation; word sense disambiguation; and many more [53, 2].

In the next section, the concept of agent-based modelling and simulation is discussed, which is another concept relevant to understanding the proposed privacy policy recommender system.

2.7 Agent-Based Modelling and Simulation

Agent-Based Modelling and Simulation (ABMS) is a novel modelling approach that is comprised of a set of autonomous, heterogeneous agents interacting with each other and their surrounding environment [45]. An *agent*, in the context of ABMS, loosely denotes any discrete, identifiable entity, with a set of identifiable properties, and a set of rules governing its behaviour [45].

Different from traditional modelling and simulation approaches, ABMS follows a bottom-up approach that enables it to simulate complex systems and phenomena. In ABMS, agents usually have a set of simple rules based on which they (i.e. agents) interact with each other and the surrounding environment. ABMS is a powerful simulation approach for simulating systems that comprise of many independent interacting components [46].

ABMS is widely used across many disciplines for a variety of applications, for example social scientists use ABMS to study the dynamics and growth of social networks, economists use it to study artificial financial markets and trade

networks, archaeologists to simulate and study the rise and decline of ancient civilizations, and biologists use it to study animal group behaviour [45, 46].

2.8 Discussion

In order to contextualise this research, topics that are of relevance to the research problem have been discussed. As SMPs are the environment within which the research problem is situated, this chapter began with an overview of the SMP, its origins, definitions, and the status quo. Next, since privacy and the preservation of privacy is the end purpose of the research, a brief discussion was conducted of the concept of privacy, and a rough definition of privacy within the context of SMPs was provided. After this, the SMP privacy landscape was surveyed along with the research done in this broad area, identifying the key research focuses, like publishing social graph data, malicious third party applications, privacy attacks in SMPs, measuring privacy in SMPs, access control models for SMPs and, of course, privacy policies automation, which is the target of the research. Through discussing these research focuses, the study highlighted the importance of privacy policies and the need for usable ways to configure those policies. Lastly, a few concepts were discussed that are of some relevance to understanding the 'nuts and bolts' of the proposed privacy policy recommender system, such as recommender systems, text classification, and agent-based modelling and simulation.

In the next chapter, a formal model of an SMP is devised. This will serve as a platform for the study's proposed privacy policy recommender system.

Chapter 3

Social Media Platform Model

3.1 Introduction

In this chapter, a model of a social media platform (SMP) is proposed, on the basis of which concepts are formally defined, such as profile attributes, user-generated content, and privacy policies. This model also serves as the platform for a proposed privacy policy recommender system.

3.2 The SMP Model

Following the footsteps of previous research [37, 23, 14], this research models SMPs as a simple undirected graph1 G=(V,E)[1] $G = (V, E)$, where V is the set of users (i.e. SMP members), and E is the set of social relationships between the users. Each node $v \in V$ represents a user, and each edge $e = (u, v) \in E$ represents a social relationship between users u and v. Figure 3.1 below, shows a simple representation of a SMP according to the above conceptualisation.

[1]In reality, many SMPs permit uni-directional relationships (e.g. *subscribe* relationships in Facebook, and the *follow* relationship in Twitter). However, for the purpose of this work, only bi-directional relationships are considered.

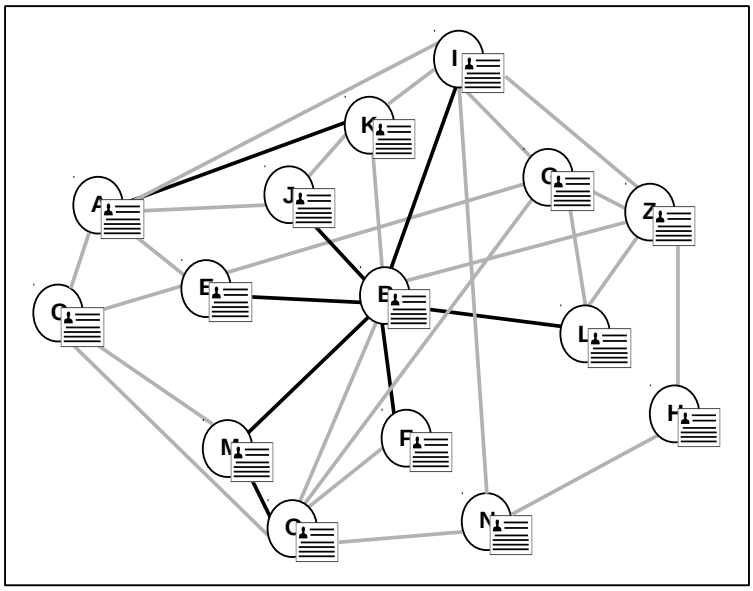


Figure 3.1: *A simple representation of a social media platform.*

3.2.1 User Profiles

SMPs allow their users to construct and maintain *profiles*, which is indeed one of the main characteristics that differentiate SMPs from other online applications. User profiles can be defined as follows:

Definition 3.1. A user profile (denoted as P) is a personal page (or space) that serves as a digital representation of the user. A user profile consists of a fixed set of *profile attributes* and a collection of *user-generated contents*.

Both profile attributes and user-generated content can be defined as follows:

Definition 3.2. Profile attributes (denoted as $A = \{a_1, a_2, \ldots, a_n\}$) are bits of information that describe the profile owner. These profile attributes may include demographic information, such as age or gender, as well as other information like email and location. Profile attributes are universal, meaning that all users have the same profile attributes, but with different values.

Definition 3.3. User-generated contents (denoted as $C = \{c_1, c_2, \ldots, c_t\}$) are, as the name indicates, pieces of content that the user generates and shares during his/her interaction in the SMP, such as posts, status updates, notes, photos,

Social Media Platform	Provide Privacy Policies?	Control Individual Attributes?	Control Individual Contents?	Allow Customised Audiences?	Privacy Policies Granularity
Facebook	✓	✓	✓	✓	Fine Grained
Google+	✓	✓	✓	✓	Fine Grained
LiveJournal	✓	✓	✓	✓	Fine Grained
Twitter	✓	✗	✗	✗	Coarse Grained
LinkedIn	✓	✓	✓	✗	Fine Grained
RenRen	✓	✗	✗	✗	Coarse Grained
ResearchGate	✓	✗	✗	✗	Coarse Grained

Table 3.1: *Privacy policies granularity level across different SMPs*

videos, etc. User-generated contents are user-specific, meaning that each user can have different contents in his/her profile.

3.2.2 Privacy Policies

Users' profiles can contain information that is quite sensitive and personal, that is why most SMPs provide users with some form of privacy policies. These privacy policies enable users to specify who can access their sensitive and personal information, thereby specifying the boundaries within which this information should reside.

The granularity (the level of flexibility and expressiveness) provided by these privacy policies varies from one SMP to another. For instance, allowing users to control access to the profile as a single unit vs. control access profile attributes individually.

Table 3.1 above shows the granularity of the privacy policies of different SMPs. The privacy policies provided by each of these SMPs were inspected and categorised into either *fine-grained* or *coarse-grained*, depending on whether or not these policies enable users to control access to individual profile attributes and individual pieces of user-generated content.

Having *fine-grained* privacy policies provides users with more detailed control over their sensitive information. Therefore, in the SMP model of this study it is assumed that the SMP provides *fine-grained* privacy policies that enable users to regulate access to every single profile attribute and piece of user-generated content in their profiles. Privacy policies can be defined as follows:

Definition 3.4. A privacy policy is a user-defined rule in the form $< item, l >$, where $item$ can be any profile attribute $a_j \in A$ or any piece of content $c_j \in C$, while $l \subseteq V$ represents the audience, that is, the set of users allowed to access the $item$. Usually users are provided with a finite set of audiences $L = \{l_1, l_2, \ldots, l_k\}$ to choose from. Some of these audiences are defined by the SMPs, while users can customise others.

Let $L = \{me, closefriends, teammates\}$ be the set of audiences that user u has. An example of a privacy policy can be something like: $< age, closefriends >$, which means, only users who are members of the *closefriends* audience are allowed to access user u's age attribute. Another example of a privacy policy is $< photo1, teammates >$, which means, only users who are members of the *teammates* audience are allowed to access $photo1$.

Privacy policies do offer SMP users a great deal of control over their personal and sensitive information, which is the essence of privacy as described in Chapter 2. However, since the SMP cannot specify ahead which privacy policies are suitable for which user, the burden of manually configuring (i.e. fine tuning) these privacy policies is left to the user's discretion. This burden causes many users to avoid using these privacy policies, and thus jeopardise their own privacy.

The next chapter discusses the theoretical framework of the privacy policy recommender system proposed to alleviate the problem of privacy policy enforcement in SMPs.

Chapter 4

Recommender System Framework

4.1 Introduction

This chapter explains the technical details and theoretical framework of the privacy policy recommender system (recommender system for short) mentioned in Chapter 1. This chapter begins with a general overview of the recommender system and then proceeds to outline its main components and functionalities.

4.2 Recommender System Overview

In order to mitigate the difficulties that many SMP users face with privacy policy configuration, a recommender system is proposed that assists the users by providing them with personalised privacy policy suggestions for both profile attributes and user-generated content, while utilising minimum user input.

The recommender system has been designed as a 'server-side' solution, with the understanding that the SMP provider will maintain it. The recommender system consists of two independent components that work in parallel to protect the users' sensitive information. The first component, termed the **Profile Attributes Protector** (PAP), relies on the privacy policies that existing users have specified for their profile attributes to suggest to presumably naïve target users

how to configure their profile attributes' privacy policies, thus minimising the amount of input required from the user. Specifically, the PAP extracts these policies from the profiles of existing experienced users and then uses this data to build several decision tree classifiers, which in turn are used to suggest suitable privacy policies to the target users for profile attributes.

The second component is termed the **User Content Protector** (UCP). The UCP learns from the target user's privacy policy history and, on the basis of this knowledge, suggests suitable privacy policies for the target user's future content. The UCP relies on the targets user's past privacy policy configurations for generated content to train a Naïve Bayes classifier that is then used to suggest privacy policies for the target user's future content.

A general overview of the proposed recommender is depicted in Figure 4.1 below. In the PAP the training data is extracted from the profiles of experienced users and used to train several decision tree classifiers. The classifiers then output privacy policy suggestions for the target user. The UCP works in a similar fashion. First, the training data is extracted from the target user's privacy policy history. This is then used to train a Naïve Bayes classifier, which outputs privacy policy suggestions for user-generated content.

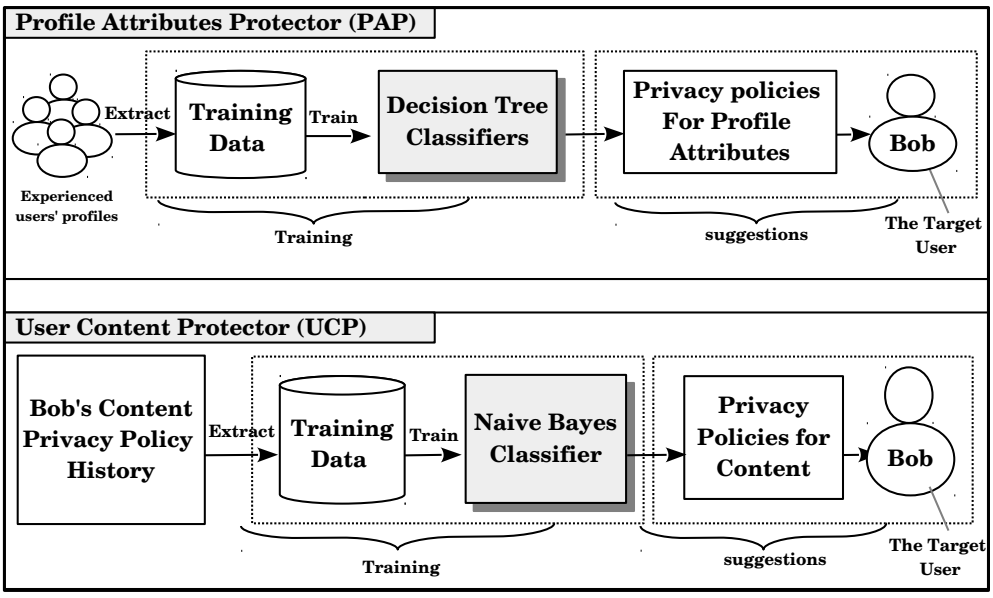


Figure 4.1: *An overview of the privacy policy recommender system.*

The following subsections describe in detail how both the PAP and UCP components, depicted in Figure 4.1 above, work to provide personalised privacy policy suggestions for users' profile attributes and content.

4.3 The Profile Attributes Protector (PAP)

The PAP is the recommender system's component responsible for protecting our target users' profile attributes. PAP is activated immediately after a new (presumably naïve) user registers at the SMP. PAP then suggests to the target user how he/she should configure the privacy policies of his/her profile attributes. However, in order to gain a deeper understanding of how the PAP works, one must first understand how it suggests privacy policies for individual profile attributes, the details of which are described in the subsection below.

4.3.1 Suggesting Privacy Policies for Individual Attributes

The general setup of suggesting privacy polices for individual profile attributes resembles traditional recommender systems described in Chapter 2. As shown in Figure 4.2, there are three classes of entities, namely *SMP users*, which correspond to the RS's users; *profile attributes*, which correspond to items; and *privacy policies*, which correspond to user-item ratings. Users are characterised by a set of demographic traits like age, education, etc.

Guided by this general setup, and in order to suggest privacy policies for profile attributes in the PAP, this study follows a *demographic-based approach* to recommender systems, where the motivating assumption is that demographically similar people have similar privacy policies for their profile attributes. The advantage of choosing a demographic-based approach over other approaches

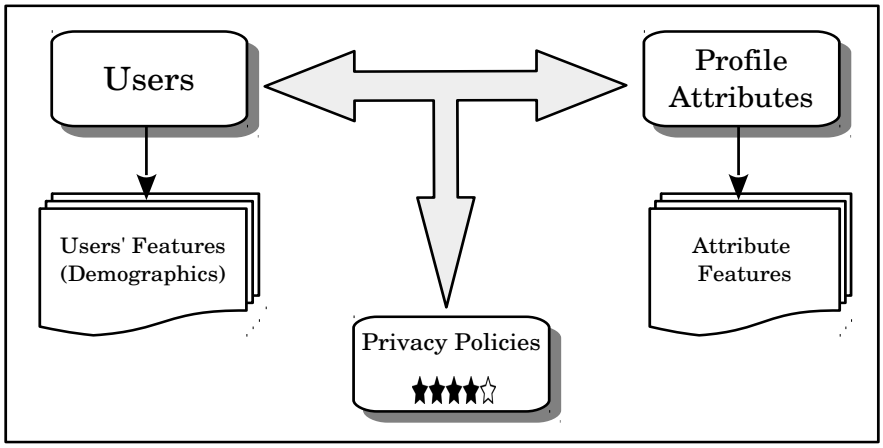


Figure 4.2: *General setup for suggesting privacy policies for profile attributes.*

(e.g. collaborative filtering), is that demographic-based approaches do not require any prior input from the target user, as they rely only on the target user's demographical information, which is already available on the target users' profile.

The process by which the PAP suggests privacy policies for an individual profile attribute say $a_i \in A$, consists of the following three phases.

4.3.1.1 Phase I: Data Collection

The first phase is the data-collection phase. In this phase, training data is collected from the profiles of existing experienced users. Since the recommender is designed to be a server-side solution, it is expected to have direct access to users' profile data. Therefore, from each existing user's profile, demographic information is extracted, as well as the privacy policy that the user (i.e. profile owner) has set for the a_i attribute. This collected information is stored in a dataset D, in which every record is of the form $(\vec{F} , l_j)$, where $\vec{F} = \langle f_1, f_2, \ldots, f_m \rangle$ is the user's demographic information, representing the features, and l_j is attribute a_i's privacy policy, and it represents the class label.

4.3.1.2 Phase II: Classifier Training

The second phase is the classifier-training phase. In this phase, a machine learning classifier is trained (i.e. built) to predict the privacy policies that the target users set for the a_i attribute. For this task a decision tree learning algorithm is selected because decision tree learning algorithms are capable of handling classification of categorical, noisy, and incomplete data [49], which are the characteristics of the training data D.

In order to train this classifier, a decision tree algorithm is applied to the dataset D, which carries out the task as follows. First, the algorithm looks at the training dataset D and finds the feature f_* that tells us the most about users' choice of privacy policies for the a_i attribute. It does this using a statistical measure called information gain (gain for short), which is given by equation (4.1) below.

$$\boxed{Gain(D, f_*) = H(D) - H(D|f_*)}$$
$$\text{where } H(x) \text{ is the entropy of } X \tag{4.1}$$
$$\text{and is given by } H(x) = -p(x)\log(p(x))$$

The algorithm then uses the feature with the highest gain (i.e. f_*) to create the root node. Next, the training dataset D is partitioned into several partitions, such that each partition contains records that have the same value for f_*. Next, for each partition p, the algorithm looks for the feature f_{*p} that has the highest gain within p and uses it (i.e. f_{*p}) to create a child node. The partition p itself is then re-partitioned and the process is repeated, until every partition has almost the same privacy policy, or the algorithm runs out of training records.

The final classifier is represented as a decision tree, wherein each tree node represents a test for a specific feature $f_i \in \vec{F}$ (i.e. demographical trait), and each

edge branching from the node corresponds to one of possible values of f_i. The leaf nodes correspond to class labels (i.e. privacy policies).

Figure 4.3 is an example of such a tree. This decision tree predicts the privacy policies of the a_i attribute, by testing the target user's features at the root node, and moving down the tree to a child node that corresponds to the value of the tested feature. The target user's features are then retested at the child node and moved down the tree progressively. The process is repeated until a leaf is reached. The privacy policy associated with this leaf is the predicted privacy policy for the a_i attribute.

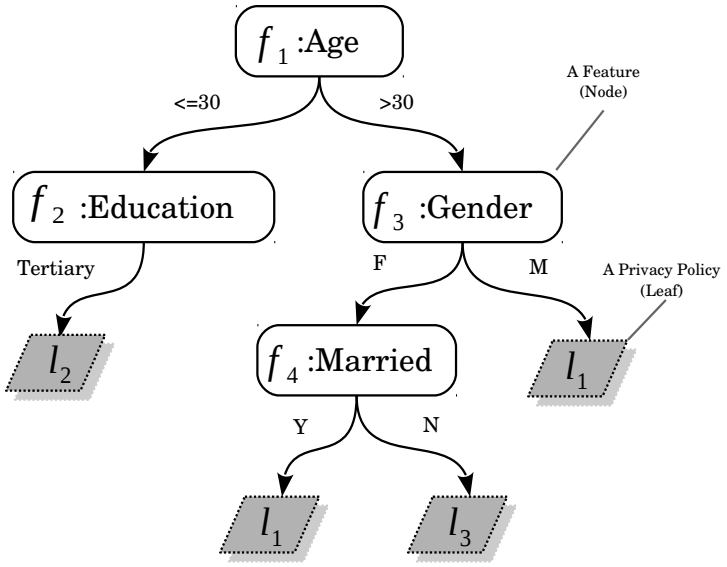


Figure 4.3: *An example of a decision tree.*

4.3.1.3 Phase III: Privacy Policy Suggestion

The third and final phase is the suggestion phase. This phase is triggered by the arrival of a target user. When a target user registers on the SMP, the features $\vec{F_{target}} = \langle f_1, f_2, \ldots, f_m \rangle$ (i.e. demographical information) are extracted form his/her newly created profile. Next, the target user's features $\vec{F_{target}} = \langle f_1, f_2, \ldots, f_m \rangle$ are passed to the decision tree classifier, which in turn predicts

a privacy policy for the a_i attribute. This privacy policy is then simply suggested to the target user. In Figure 4.4 below, the process of suggesting privacy policies for individual profile attributes is depicted, as described above.

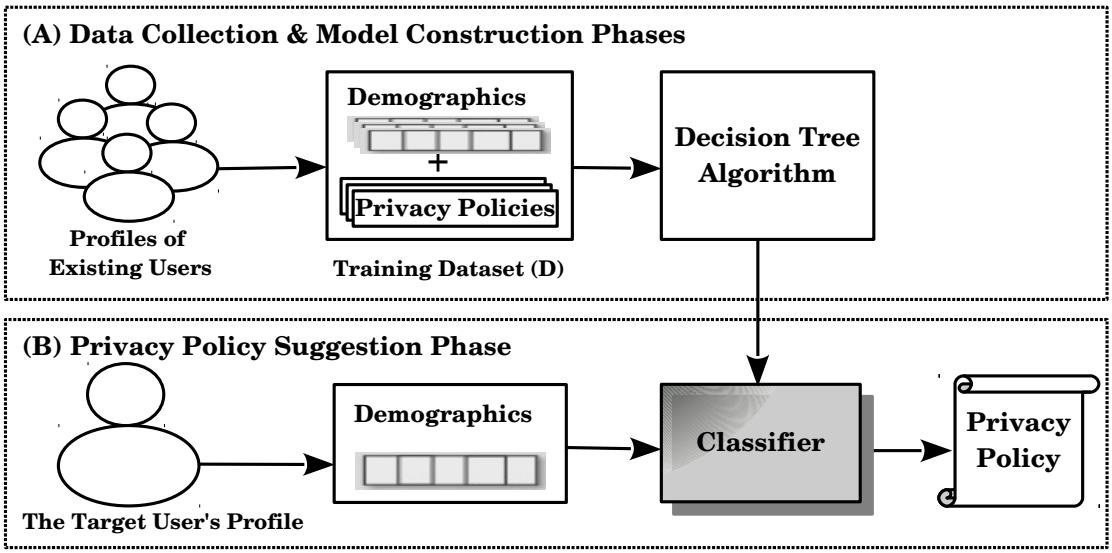


Figure 4.4: *Suggesting privacy policies for individual profile attributes.*

4.3.2 Suggesting Privacy Policies For All Attributes

So far, a description has been given of how privacy policies are suggested for one profile attribute. In order to suggest privacy policies for all the attributes in the target user's profile, the previously described process is repeated for each one of the profile attributes.

Specifically, as shown in Figure Figure 4.5, demographic information and privacy policies are extracted from the profiles of existing users. This data is then used to build a series of training datasets $\{D_1, D_2, \ldots, D_{|A|}\}$, one for each profile attribute $a_i \in A$. Afterwards, the decision tree learning algorithm is applied to these training datasets, to train a series of classifiers $CL = \{cl_1, cl_2, \ldots, cl_{|A|}\}$. This is such that each classifier $cl_i \in CL$ predicts the privacy policies that the target users will set for an attribute $a_i \in A,\ \forall i \in \{1, 2, \ldots, |A|\}$. Finally, each

time a target user registers at the SMP, his/her demographics are passed to every classifier $cl_i \in CL$, to predict the privacy policies of every attribute in the target user's profile. These predicted policies are then suggested to the target user.

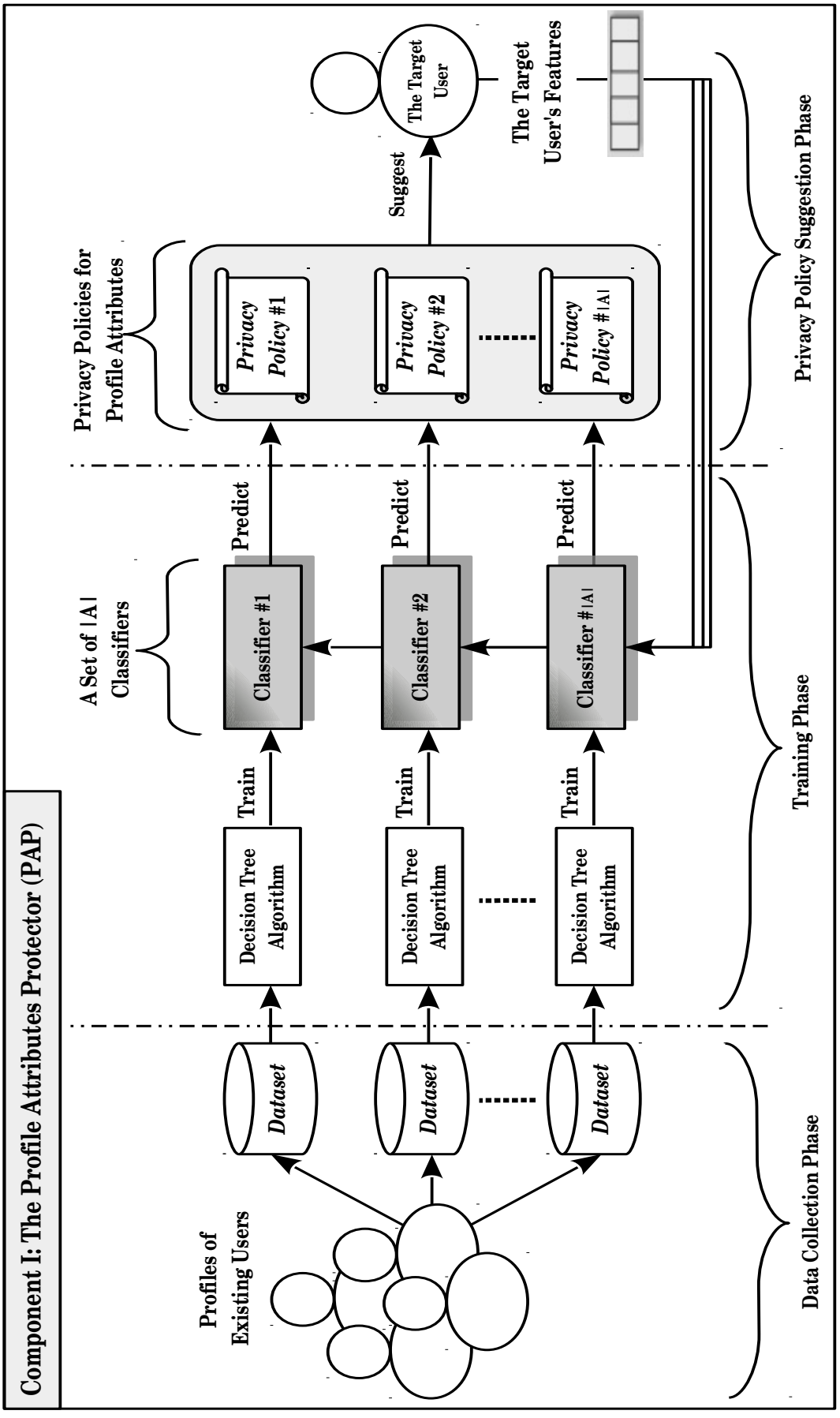


Figure 4.5: Component I, the process of suggesting privacy policies for profile attributes as a whole.

The following section describes in detail how the second component of the recommender system (i.e. UCP) works.

4.4 The User Content Protector (UCP)

The UCP is the component responsible for protecting our target users' content. As mentioned earlier, UCP is activated after the target user joins the SMP, where it 'learns' from the privacy policies that the target user has specified for some of his/her previous content, and uses this knowledge to suggest privacy policies for the target user's future content.

The general setup for suggesting privacy policies for user-generated content (which is depicted in Figure 4.6 below), consists of three main classes of entities, namely *SMPs users*, which represent the RS's users; *user-generated content*, which represent items; and the *privacy policies* that the target user specifies for his/her content, and which represent user-item ratings. Each piece of user-generated content can be characterised by a set of features.

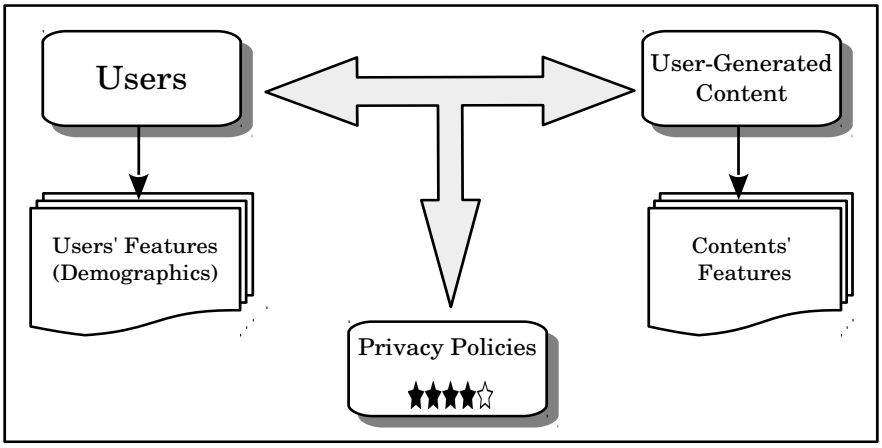


Figure 4.6: *General setup for suggesting privacy policies for user-generated.*

Guided by this general setup, the UCP follows a *content-based approach* to recommender systems, where the motivating assumption is that similar pieces

of user-generated content have similar privacy policies. A content-based approach was selected for UCP because usually each user generates different content to others. Therefore, its not feasible to suggest privacy policies for content generated by one user; based on the privacy policies that other users have specified for their content, like in the case of collaborative filtering, and demographic-based approach recommender systems.

The process by which the UCP suggests privacy policies for content generated by a particular target user, say $u_t \in V$, consists of the following four phases.

4.4.1.1 Phase I: Data Collection

The first phase is the data-collection phase. In this phase the privacy policy history of the target user's content is collected. More specifically, this is all content generated by the target user, in addition to the privacy policy associated with that content. This collected data is stored in a dataset Ω, in which every record is in the form (c_i, l_j), where c_i is a piece of content generated by the target user u_t, and l_j is the privacy policy that u_t has specified for c_i.

4.4.1.2 Phase II: Preprocessing of The Data

The second phase is the preprocessing phase. In this phase every piece of content $c_i \in \Omega$, is transformed to a vector of features that characterise c_i, thereby transforming the row dataset Ω into a dataset of labelled feature vectors $\hat{\Omega}$, in which every record is in the form $(\vec{c_i} , l_j)$, where, $\vec{c_i} = \langle f_{i1}, f_{i2}, \ldots, f_{im} \rangle$ is c_i's feature vector. The preprocessing phase can be implemented for all types of user-generated content; however, the features might differ from one content type to another. For instance, in text-based content like status updates, features might be words; while in visual content like photos, features may include metadata, tagged persons, whether the photo contains faces, and so on.

4.4.1.3 Phase III: Classifier Training

The third phase is the classifier-training phase. In this phase the classifiers that predict privacy policies for the target user's future content are trained. This is such that one classifier is trained for each type of user-generated content. Unlike the PAP profile attribute classifiers, these content classifiers are unique to the target user whose privacy history they (i.e. the classifiers) were trained with.

For constructing these classifiers, the Naïve Bayes classification algorithm is used, which is a member of a well-known family of classification algorithms that are based on Bayes theorem (given in equation (4.2)).

$$\boxed{P(B|A) = \frac{P(A|B)P(B)}{P(A)}} \tag{4.2}$$

The Naïve Bayes was chosen over the decision trees algorithm because it is more suitable for the task, as it is quicker to train and test, which is important for predicting privacy policies for user-generated content, where the time between creating and sharing a content is very short.

In order to build the classifier, the Naïve Bayes algorithm is applied to the pre-processed dataset $\hat{\Omega}$. Naïve Bayes follows a probabilistic model whereby, in order to predict the privacy policy for a piece of user-generated content c_i, Naïve Bayes selects the privacy policy $l_* \in L$ that maximises the probability $P(l_j|f_1, f_2, \ldots, f_m) \quad \forall l_j \in L$, which is the probability that the privacy policy l_j will be observed given that the features $\{f_1, f_2, \ldots, f_m\}$ that characterise content c_i are observed. This privacy policy (i.e. l_*) is determined as follows.

First, according to Bayes theorem (given by equation (4.2) above), the probability that the privacy policy l_j will be observed given that c_i's features (i.e.

$\{f_1, f_2, \ldots, f_m\}$) are observed, is calculated as follows.

$$P(l_j|f_1, f_2, \ldots, f_m) = \frac{P(f_1, f_2, \ldots, f_m|l_j)P(l_j)}{P(f_1, f_2, \ldots, f_m)} \tag{4.3}$$

Now, by assuming that the c_i's features are conditionally independent[1], equation (4.3) above can be rewritten as:

$$P(l_j|f_1, f_2, \ldots, f_m) = \frac{\left(\prod_{i=1}^{m} P(f_i|l_j)\right)P(l_j)}{P(f_1, f_2, \ldots, f_m)} \tag{4.4}$$

By definition, l_* is the privacy policy that maximises the probability $P(l_j|f_1, f_2, \ldots, f_m)$, which, according equation (4.4) above, is the privacy policy that maximises the following term.

$$l_* = \arg\max_{l_j \in L} \left[\frac{\left(\prod_{i=1}^{m} P(f_i|l_j)\right)P(l_j)}{P(f_1, f_2, \ldots, f_m)}\right] \tag{4.5}$$

Since $P(f_1, f_2, \ldots, f_m)$ is constant for all $l_j \in L$, then

$$\boxed{l_* = \arg\max_{l_j \in L} \left[\left(\prod_{i=1}^{m} P(f_i|l_j)\right)P(l_j)\right]} \tag{4.6}$$

Where the probability $P(l_j)$ and each of the conditional probabilities $P(f_i|l_j)$ are estimated from the preprocessed dataset $\hat{\Omega}$.

4.4.1.4 Phase IV: Privacy Policy Suggestion

The fourth and final phase is the suggestion phase. This phase is triggered when the target user u_t generates any new piece of content c_{new}. This content (i.e. c_{new}) is first preprocessed to transform it into a vector of features. This feature vector is then passed to u_t's classifier, which in turn predicts c_{new}'s privacy policy. Following this, the predicted policy is simply suggested to u_t. Figure 4.7

[1]This assumption is referred to as the *naïve assumption*, hence the name *naïve bayes*. It is called naïve assumption because it rarely holds in real life.

below depicts the process of suggesting privacy policies for user-generated content.

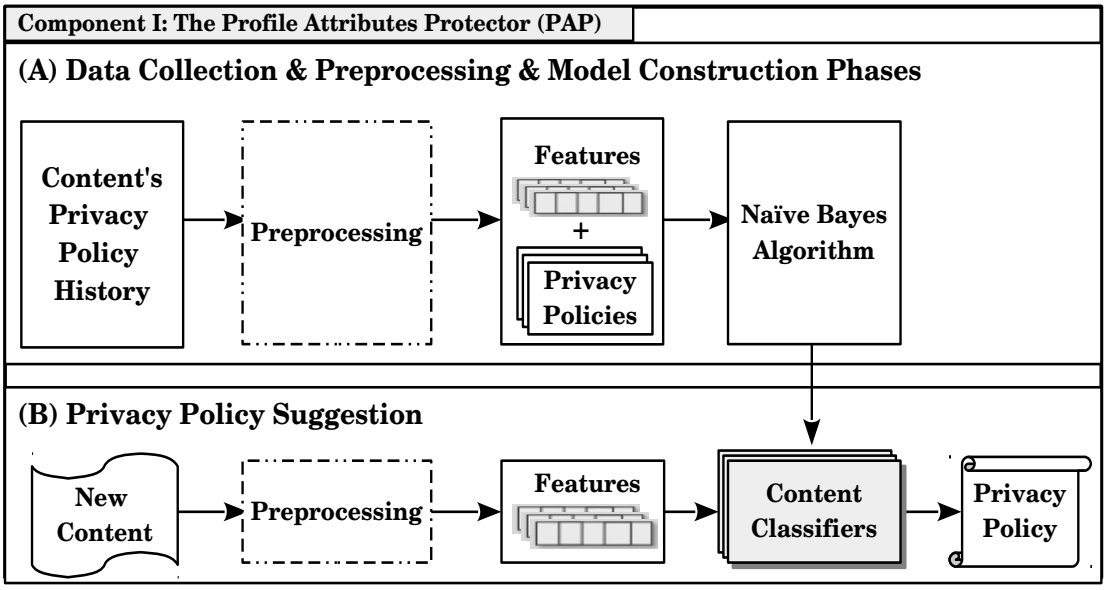


Figure 4.7: *The process of suggesting privacy policies for user-generated content.*

4.5 Privacy Analysis

Privacy policies enable users to protect their privacy by allowing users to specify the boundaries within which their sensitive information resides. However, since these boundaries might differ from one user to another, SMP providers tend to provide open (i.e. very permissive) default privacy policies, thus leaving the task of configuring (i.e. fine tuning) these policies to its users. As a consequence, by default new SMP users have no, or at best very little, privacy, unless they configure their privacy policies to suit their privacy needs.

However, for 'privacy-aware' users who invest the time and effort to configure their privacy policies, it is assumed that their policies are correct, and more privacy preserving of the SMP's default policies.

Therefore, by relying on the privacy policies of the existing SMP users, the suggested recommender system ensures that targets users' profile attribute privacy policies are as privacy-preserving as the policies of existing 'privacy-aware' users, which is better than the SMP's default and, by relying on the target user's privacy history, the recommender system ensures that target user's content's privacy policies are as privacy-preserving as the user's own privacy history, which is also better than the SMP's default.

The next chapter discusses an experiment carried out by the study for implementing a basic prototype of the privacy policy recommender system, as well as the results obtained from these experiments.

Chapter 5

Implementation and Results

5.1 Introduction

Building upon the privacy policy recommender system framework theorised in Chapter 4, this chapter walks through the design choices and the experiments conducted to implement a basic prototype of the proposed privacy policy recommender system. In addition to the results obtained from these experiments.

This chapter begins by describing the experimental platforms used for implementation..

5.2 The Experimental Platforms

In order to implement the privacy policy recommender system prototype, several software tools and packages were used, including the following:

1. **Facebook Graph API**[1]**:** This is an application programming interface developed by Facebook. This Graph API enables third-party (i.e. external) developers to connect to Facebook's social graph and gain 'regulated' access to users' information. Facebook Graph API was used to collect some

[1]Documentation: https://developers.facebook.com/docs/graph-api

of the SMP data required for experiments, as described later in this chapter.

2. **NetLogo**[2]**:** A widely used agent-based modelling and simulation (ABMS) environment, developed at Northwestern University [69]. NetLogo is used to synthetically generate the SMP data required for experiments, as described later in this chapter. Figure 5.1 below shows a snapshot of the NetLogo's interface.

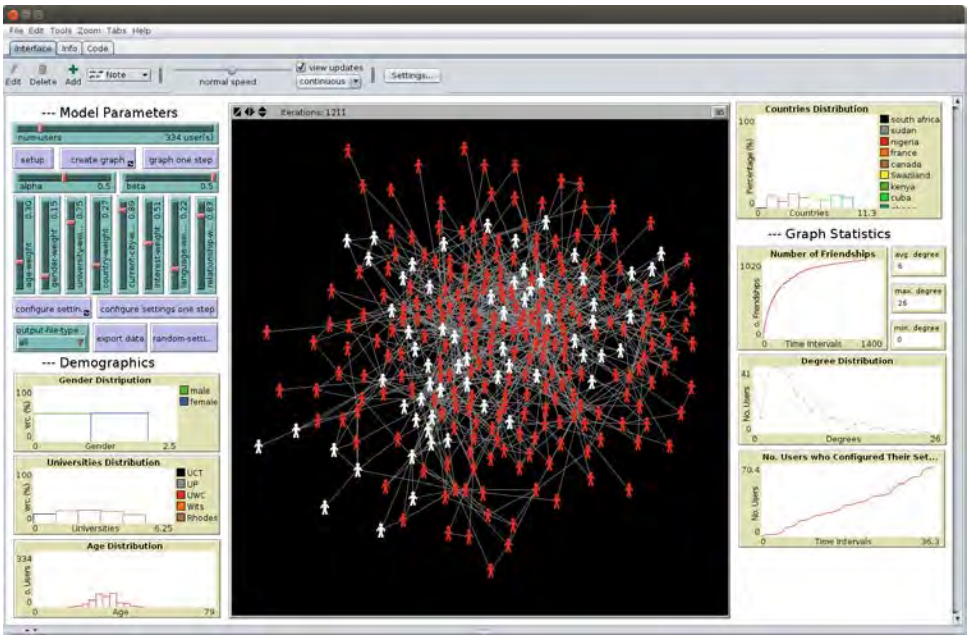


Figure 5.1: *A snapshot of NetLogo, featuring our SMP simulation model.*

3. **Weka**[3]**:** A popular machine learning/data mining package developed at the University of Waikato. Weka provides a variety of algorithms for data preprocessing, classification, clustering, association, and visualisation [34]. Weka's algorithm implementations were used to train and validate the recommender system's classifiers, as described later in this chapter. Figure 5.2 shows a snapshot of Weka's interface.

[2]Website: https://ccl.northwestern.edu/netlogo/
[3]Website: http://www.cs.waikato.ac.nz/ml/weka/

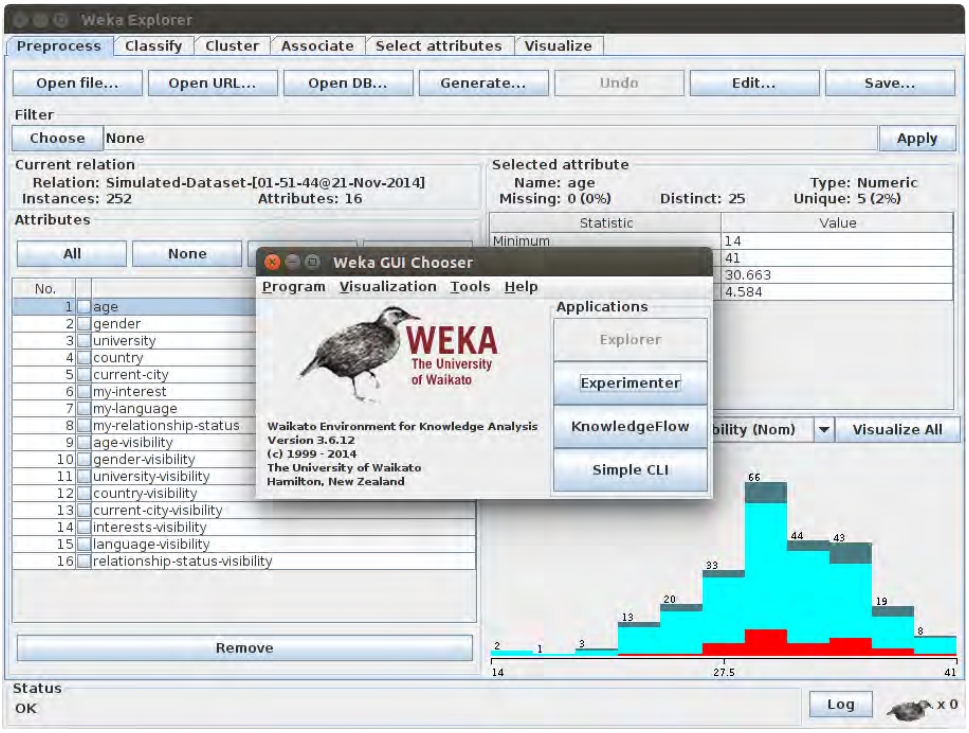


Figure 5.2: *Weka machine learning platform.*

4. **Experimental Workstation:** All of the experiments were conducted on a workstation, with an Intel core i7-4790 3.60 GHz processor, and 8GB of RAM.

Having described the experimental platforms, the following sections provide the details of implementing the recommender system's prototype and the obtained results.

5.3 Implementing The Recommender System

In order to implement the proposed privacy policy recommender system, several experiments were carried out that were primarily focused on building the classifiers that predict privacy policies for target users. These experiments include both the profile attributes protector (PAP), as well as the user content protector (UCP).

5.3.1 Implementing The Profile Attributes Protector (PAP)

In order to implement the profile attributes protector (PAP), focus was placed on building the most important elements of this component, which are the classifiers that predict the privacy policies of the target users' profile attributes. To start, the necessary training data for building these classifiers was collected.

5.3.1.1 Data Collection

In order to train (i.e. build) the PAP profile attributes classifiers, it is essential to obtain rich datasets of users' demographic information and privacy policies. However, acquiring such rich datasets is difficult due to the sensitive nature of SMPs' data, and the numerous privacy and ethical issues associated with it [73]. As an alternative, the researchers opted to synthetically generate training datasets through simulating an SMP.

At later stages of the work, a real dataset was obtained (henceforth, referred to as the CFPRS dataset) that was used in a similar study by Alsalibi and Zakaria [4]. Even though the CFPRS dataset is structurally different to the simulated datasets, it can provide a valuable insight as to how the recommender will behave with real data. Therefore, it was decided to use both the simulated and the CFPRS datasets during experiments.

The following sub-sections describe how synthetic datasets were generated through simulation, and also briefly describe the CFPRS dataset.

5.3.1.1.1 The SMP Simulation Model

In order to generate relatively realistic data to train the PAP classifiers, the researchers opted to simulate an SMP. In order to simulate the SMP, the Agent-Based Modelling and Simulation approach (ABMS) was followed, as described in Chapter 2. An ABMS approach was selected because SMPs are complex systems formed by many users interacting with each other and, according to Macal [46], ABMS is suitable for simulating such complex systems. The SMP simulation was implemented using the NetLogo ABMS environment.

The main building blocks of the SMP simulation are agents, which represent the SMP's users. For simplicity, each agent is assigned only eight profile attributes: $A = \{age, gender, university, country, city, interests, language, relationship\ status\}$. In addition to profile attributes, each agent is characterised by three internal properties that govern its behaviour, namely a friendship threshold $\lambda \in [0, 1]$, which indicates how friendly an agent is; a maximum nodal degree δ, which represents the maximum number of friendships an agent can maintain [6]; and lastly, a privacy concern $\theta \in [0, 1]$, which quantifies the agent's level of privacy concern [33].

The values of the agents' categorical attributes are given values that are selected uniformly from a predefined value-set. It is assumed that the numerical *age* attribute is normally distributed. As for the internal properties, it is assumed that the values of the *friendship threshold*, the *maximum nodal degree*, and the *privacy concern* are also drown from a normal distribution.

In order to simulate how a user's friendship graph is formed, two concepts are relied on, namely homophily and triadic closures. Homophily is the love of the same [8], while triadic closures describe the tendency to make friendships with friends of friends [39]. The idea of using these concepts is to ensure that the

SMP follows the same pattern of friendship formation as is the case in typical real-life SMPs. The friendship graph is formed as follows: each agent u checks whether its current number of friends has reached its internal maximum nodal degree δ_u. If this is not the case, then u selects another random agent v, and calculates the percentage of their identical profile attributes $S_{(u,v)} \in [0, 1]$ (given by equation (5.1) below), and percentage of their mutual friends $mf_{(u,v)} \in [0, 1]$ (given by equation (5.2) below).

$$\boxed{S_{(u,v)} = \sum_{j=1}^{|A|} \left(\frac{\mathbb{1}\{a_{uj} = a_{vj}\}}{|A|} \right)} \tag{5.1}$$

Where, u , v are agents and a_{uj}, a_{vj} are their corresponding profile attributes. While $\mathbb{1}\{condition\}$ is an indicator function that returns 1 when $condition$ is true, and 0 otherwise.

$$\boxed{mf_{(u,v)} = \left(\frac{mutual(u, v)}{friends(u)} \right)} \tag{5.2}$$

Where, $mutual(u, v)$ is the number of common friends between agents u & v, and $friends(x)$ is the number of friends agent x has.

Next, if the friendship score given by a linear combination of $S_{(u,v)}$ and $mf_{(u,v)}$, is greater than u's internal friendship threshold λ_u, then a friendship is formed between u and v. This 'socialisation' process is repeated until all agents reach their maximum nodal degrees (i.e. no. friendships).

The privacy policy configuration process is inspired by the work of Guo and Chen [33], in which profile attributes are classified by sensitivity (that is privacy weights), and the privacy policies that are assigned for an attribute are influenced by the user's level of privacy concern and the attribute's privacy weight. Following this heuristic, each profile attribute $a_j \in A$ was assigned, an arbitrary privacy weight $w_{a_j} \in [0, 1]$ that indicates a_j's sensitivity. Next, in

order to set a privacy policy for an attribute, say a_i, each agent u was modelled to first calculate a_i's privacy score ($score_{a_i}$), which is a linear combination of u's privacy concern θ_u, and a_i's privacy weight w_{a_i}. The numerical privacy score is then mapped to one of four possible audiences: $L = \{$only me, friends, friends of friends, public$\}$. Such that the lower the privacy score, the wider the audience is (e.g. $score_{a_i} = 0 \Longrightarrow l = public$).

The SMP simulation model was designed to be flexible, where the number of agents, and the weights of profile attributes are parameters that can be modified at every simulation. This simulation model also facilitates exporting the data generated by each simulation in several standard formats like *arff* and *csv*. The full implementation details of the simulation can be found in Appendix A.

5.3.1.1.2 Generating Synthetic Training Datasets

The above SMP simulation model was used to generate synthetic training datasets. Specifically, the model was used to run several simulations of SMPs of different sizes. Then, by the end of each of these simulations, every agent's demographic data and profile attributes' privacy policies were collected and stored in a dataset. Several control datasets are also generated, where the privacy policies are assigned randomly to the agent's demographics. The reason behind this is to make sure that whatever patterns observed in the simulated datasets (if any) are not products of mere randomness.

5.3.1.1.3 The CFPRS Dataset

The CFPRS dataset was used in a similar study by Alsalibi and Zakaria [4]. Alsalibi and Zakaria collected the data via an online survey, where they prompted the 477 respondents for the values of eleven profile attributes, namely user name, date of birth, location, gender, relationship status, education, religion,

email, cellphone, languages and profile picture. Alsalibi and Zakaria then asked the respondents to specify suitable privacy policies for these profile attributes by choosing from four possible audiences, namely Public, Friends, Friends of Friends, and Only Me. However, for this study the researchers were granted access to only a subset of the original dataset. The final dataset contains the actual values of three attributes only, namely gender, location, and religion attributes, in addition to the privacy policies of all eleven attributes.

Now that both of the training datasets have been described, the next sub-section discusses how the PAP's classifiers are built and validated.

5.3.1.2 Classifiers Training And Evaluation

In order to train (i.e. build) the PAP profile attributes' classifiers, the J48 algorithm was used, which is Weka's implementation of the C4.5 decision tree learning algorithm. Specifically, the J48 algorithm was applied to the synthetic datasets (both simulated and random) of the research in order to train eight classifiers, one for every profile attribute in the SMP simulation model. Each of these classifiers was trained several times, using datasets of different sizes each time.

Next, in order to measure how well these classifiers can actually predict the privacy policies of the target users' profile attributes, an evaluation method was used called n-fold *cross-validation*. In the n-fold *cross-validation* method, the training dataset is divided into n equal parts, such that $n - 1$ parts are used to build a classifier. This classifier is then used to predict the privacy policies for the remaining part. The privacy policies predicted by this classifier are then compared against the already known privacy policies for that part. This process is repeated n times, and the best performing classifier is reported.

In this case, a 10-fold cross validation test (Weka's default) was performed on the classifiers, recording the *accuracy* of each classifier, that is the percentage of the records that the classifier was able to predict the privacy policies for correctly. The cross-validation results showed a significant difference between simulated and random datasets. The classifiers trained on simulated datasets achieved an accuracy ranging from 60% to 80%, while the classifiers trained on random datasets achieved an accuracy ranging from 20% to 30%. This is evident in Figure 5.3 below, which shows the accuracy of each profile attribute's classifier, plotted against the size and the type of its training dataset.

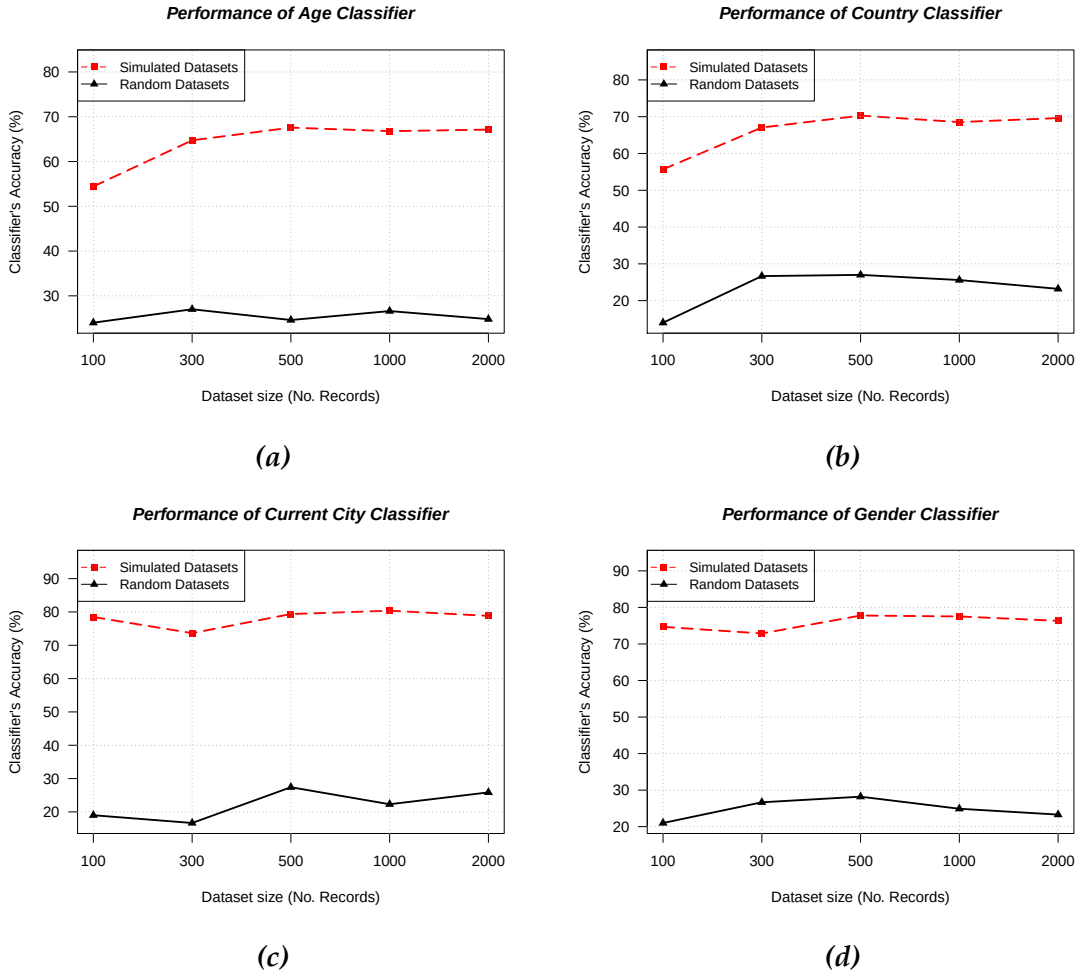


Figure 5.3: *(figure continued in the next page.)*

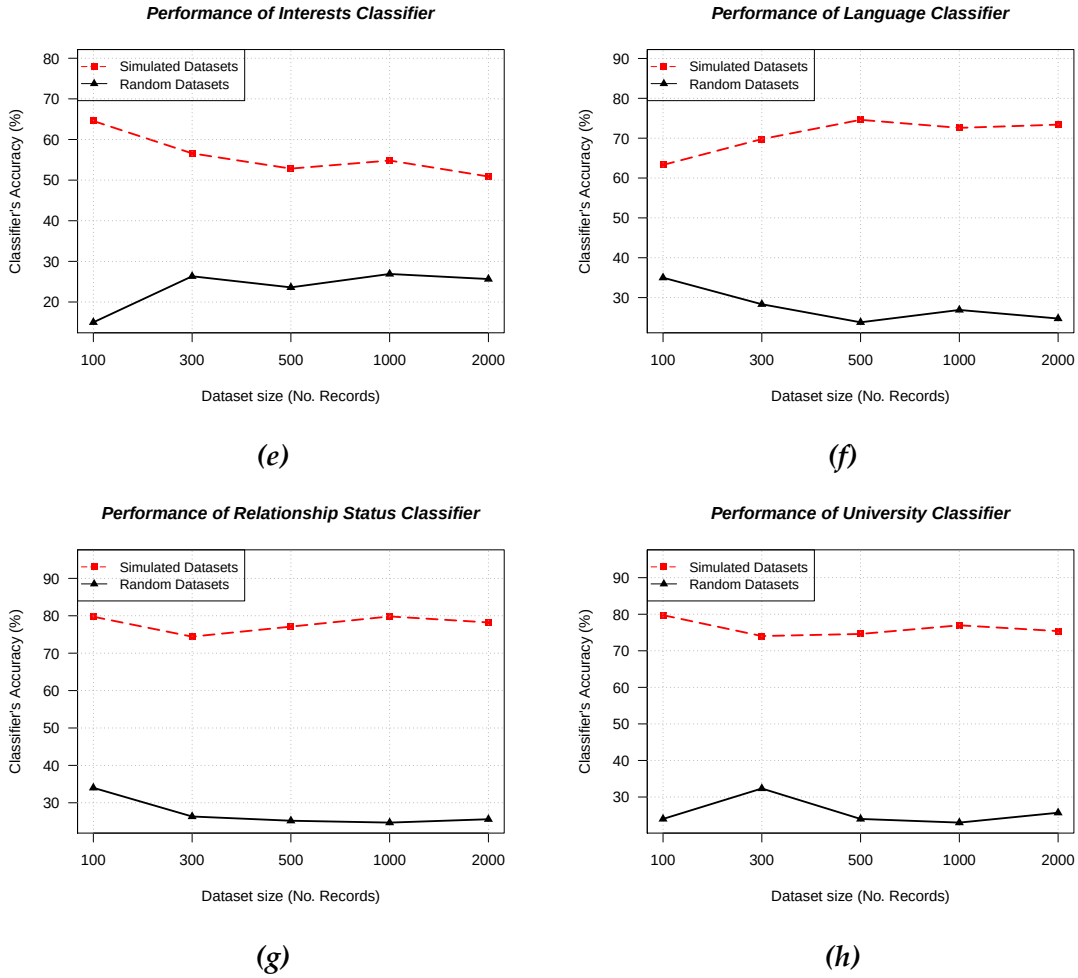
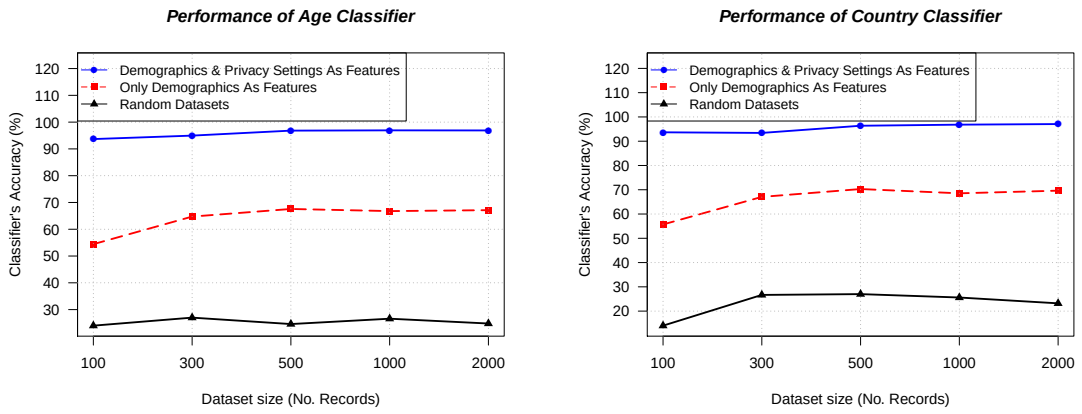


Figure 5.3: *The performance of the classifier of every profile attribute, when trained on both random and simulated datasets. Such that, the accuracy of each classifier is plotted against the size of the dataset used for training it.*

In order to further enhance the accuracy of the classifiers, the structure of the training datasets was changed. Specifically, the features used in training the classifiers were extended by incorporating the privacy policies of other profile attributes into the feature vector. For example, to train the classifier of the *age* attribute, features were used that are a combination of users' demographic information and the privacy policies they set for other profile attributes. That is, the privacy policies of other profile attributes in $\{A - \{age\}\}$.

Classifiers were then retrained on the new datasets, and the accuracy of each classifier was measured using the aforementioned methodologies. The results showed a noticeable 15% to 25% improvement in the accuracy of the classifiers trained on simulated datasets. On the other hand, no significant improvement was noted on the accuracy of classifiers trained on random datasets. This is evident in Figure 5.4 below, which shows the accuracy of each profile attributes classifier, plotted against the size and the type of the dataset used for training that particular classifier.



(a) ***(b)***

Figure 5.4: *(figure continued in the next page.)*

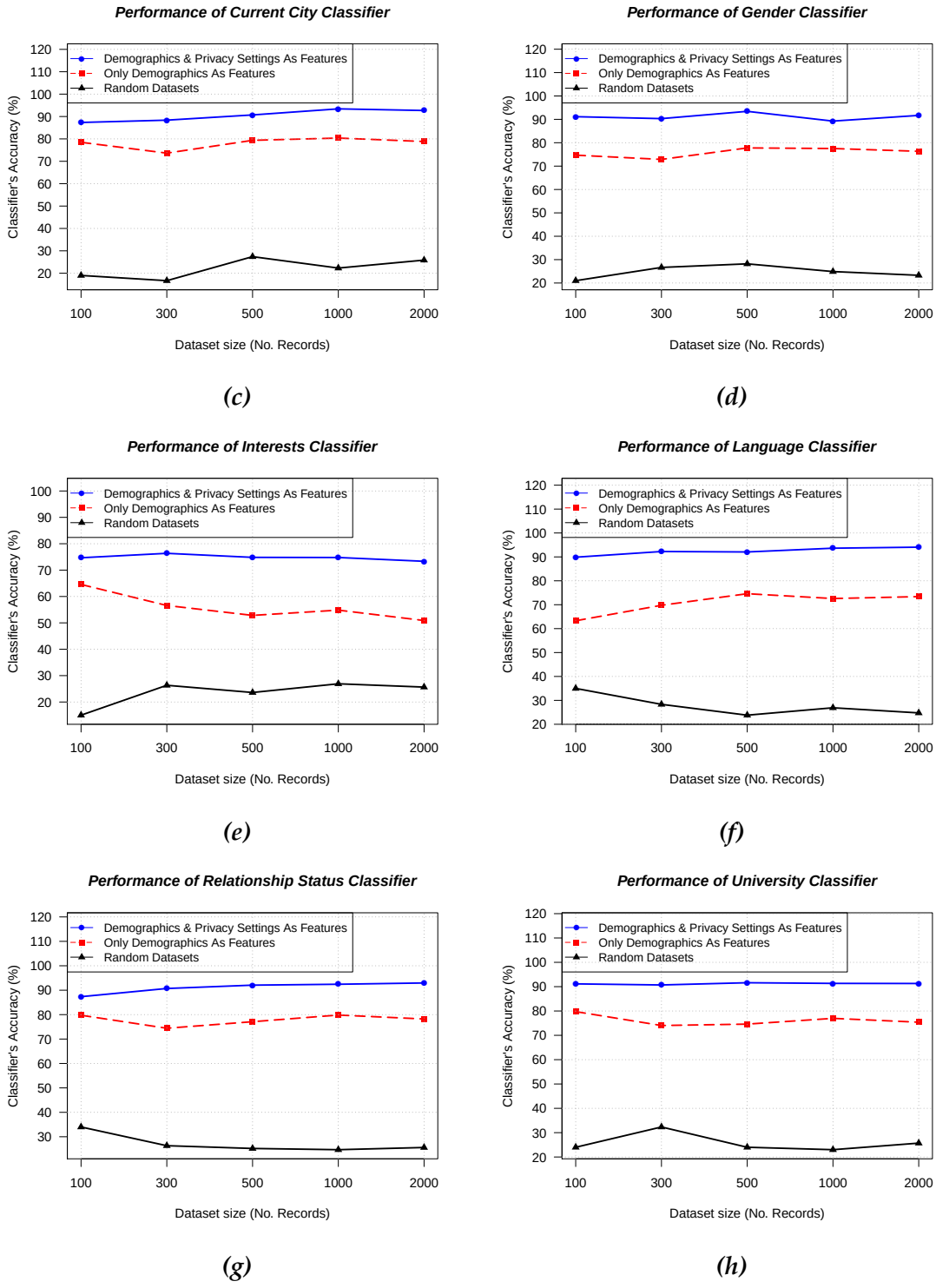
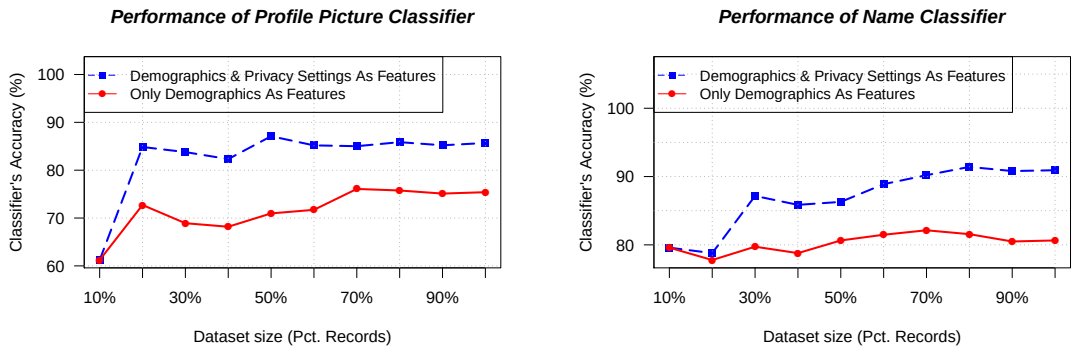


Figure 5.4: *The performance of the classifier of every profile attribute, when trained on simulated vs. random datasets, while using extended vs. only demographic features. Such that, the accuracy of each classifier is plotted against the size of the dataset used for training it.*

In addition to synthetic datasets, experiments were also conducted training the PAP classifiers on a real dataset. The J48 algorithm was applied to the CFPRS dataset to train eleven classifiers, one for each profile attribute in the CFPRS dataset. Each of these classifiers is trained several times, such that each time an increasing percentage of records were used from the CFPRS dataset.

Next, in order to evaluate the performance of these classifiers, a 10-fold cross validation test carried out, and the accuracy of each classifier was recorded as it was being trained using different percentage of records.

The cross validation results showed that the accuracy of the classifiers ranged on average from 60% to 70%, which is a bit lower than the classifiers trained on the simulated datasets. However, similar to the simulated datasets, a noticeable average of 17% improvement was observed in the classifiers' performance when demographics and privacy policies were used as features. This is depicted in Figure 5.6 below, which shows the accuracy of the classifiers trained on the CFPRS dataset, where the accuracy of each classifier is plotted against the percentage of CFPRS records used to train it.



(a) *(b)*

Figure 5.5: *(figure continued in the next page.)*

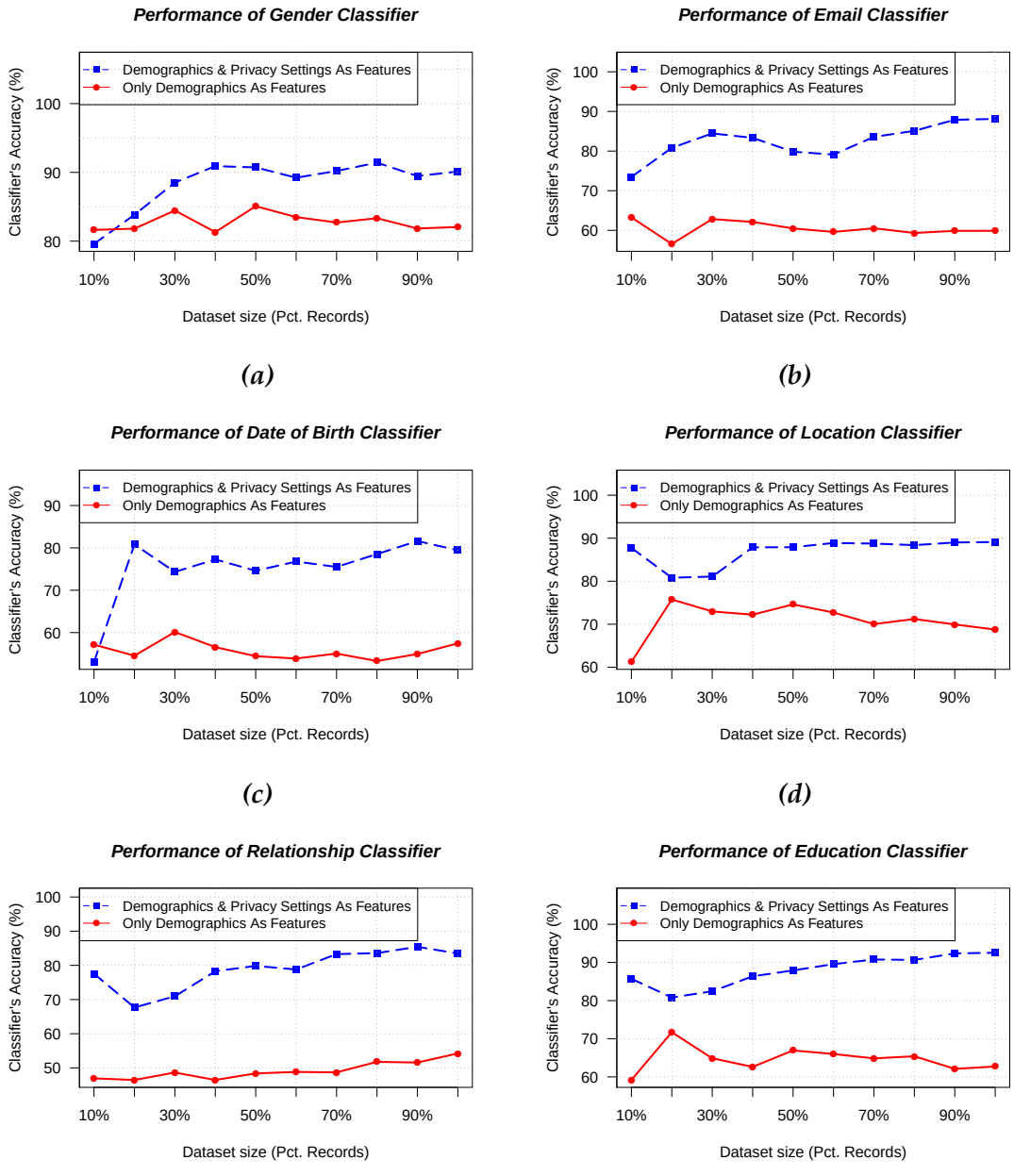


Figure 5.6: *(figure continued in the next page.)*

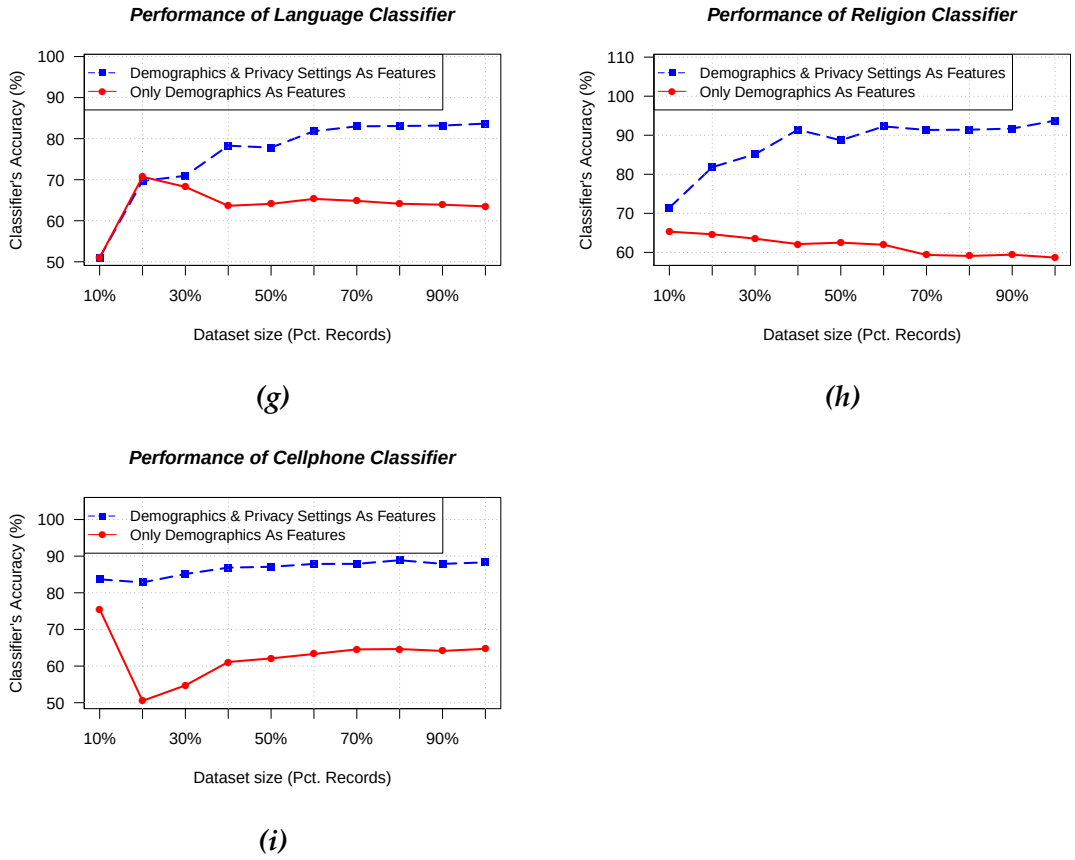


Figure 5.6: *The performance of the PAP profile attributes classifiers as they trained on the CFPRS dataset, while using extended vs. only demographic features. The classifiers were trained multiple times on different percentage of record from the CFPRS dataset, where, the accuracy of each classifier is plotted against the percentage of the dataset records used for training it.*

In the following section the experiments carried out for implementing the user content protector (UCP) component are reviewed.

5.3.2 Implementing The User Content Protector (UCP)

In order to implement the user content protector (UCP) component, the research focused on building the classifiers that predict the privacy policies for targeted user's content because they are the most critical element of the UCP.

However, despite the fact that target users can generate/share different types of content, for the purpose of these experiments, only text-based content was worked with because of its availability. As a consequence, the study focuses on building one classifier, for one type of user-generated content (i.e. text-based content). Nonetheless, it is still possible to do so with other content types, like photos for instance [57]. Next, the necessary data is collected for building (i.e. training) the classifier.

5.3.2.1 Data Collection

In order to collect training data Ω, the researchers used Facebook's graph application programming interface (API) to download textual posts from the author's Facebook account, which will serve as text-based user-generated content. Particularly, from each post three fields are recorded: *message* which holds the actual content of the post, *description* which contains descriptions of hyperlinks within the post (if any), and finally, *name* which contains the name of the page or the person the post is being shared from.

Next, since all of the downloaded posts were shared under the default privacy policy, their privacy policies were manually configured. Each post was assigned a privacy policy by selecting from one of three audiences, namely

$L = \{allfriends, closefriends, liberals\}$. The general guideline followed in labelling posts is: if the post is personal it should be visible to *closefriends*, if controversial it should be visible to *liberals*, otherwise it should be visible to *all friends*.

In total, 596 posts were downloaded, 293 of them were written in English, and 290 were written in Arabic, while 13 posts were written in both. In the next step the training data Ω is preprocessed, so that it could be used by text classification algorithms to train the PAP content classifier.

5.3.2.2 Preprocessing The Data

Since the study is limited to text-based content, the *text classification* preprocessing methods that described in Chapter 2 were used. Specifically, in order to preprocess the *corpus* of textual posts Ω, Weka's *StringToWordVector* filter was used. The StringToWordVector filter provides many options for preprocessing steps like tokenisation, stemming, normalisation, etc.

The preprocessing phase begins by normalising each content (i.e. post) $c_i \in \Omega$ by lower-casing letters and removing diacritical marks[4]. Then, the data was stemmed using the *lovins stemmer* provided by the StringToWordVector filter[5]. Next, to create the term vocabulary $\mathbb{V}$, word tokenisation was used, whereby every content $c_i \in \Omega$ is broken down into words (i.e. blocks of consecutive characters). However, only a maximum of 100 words were kept (Weka's default) from each class in the vocabulary to reduce the dimensionality of the resulting feature vectors. Lastly, for vectorisation the study followed a binary approach, where only the terms existence in content was registered, with no regard for its frequency.

[4]For normalising Arabic content, the Ar-PHP package https://github.com/tawfekov/ar-php was used

[5]Since the *lovins stemmer* only works on English content, Arabic content was left unstemmed.

After the above preprocessing steps, the raw textual dataset Ω is transformed into a dataset of labelled feature vectors $\hat{\Omega}$, ready to be used to build the content's classifier, as discussed in the following section.

5.3.2.3 The Classifier Training and Validation

Before the UCP content classifier is trained, the preprocessed dataset $\hat{\Omega}$ is first divided into a *training & validation* dataset TV, which contains 70% of the records in the preprocessed dataset; and a test *test* dataset Te that contains the remaining 30%. This is as usually advised in text classification tasks [53].

Next, for building the UCP content classifier, the researchers experimented with three variations of the Naïve Bayes algorithm, namely *Naïve Bayes*, *multinomial Naïve Bayes*, and *complement Naïve Bayes*, in addition to Weka's *Sequential Minimal Optimization (SMO)* algorithm. Each of these algorithms was applied on the training & validation dataset TV, which resulted in the construction of several classifiers.

However, since only one classifier is needed to predict the privacy policies of the target user's content, a way was required to select one of these classifiers to be the main PAP content classifier. Therefore, a 10-fold cross-validation test was performed with the intention of selecting the best performing classifier.

The cross-validation results show that the classifiers' accuracy ranged between 45%-50%, and the best performing classifier was the one trained using *complement Naïve Bayes*, which achieved 50.11% accuracy. Figure 5.7 below shows the accuracy of the classifiers resulting from applying the above classification algorithms to the TV dataset.

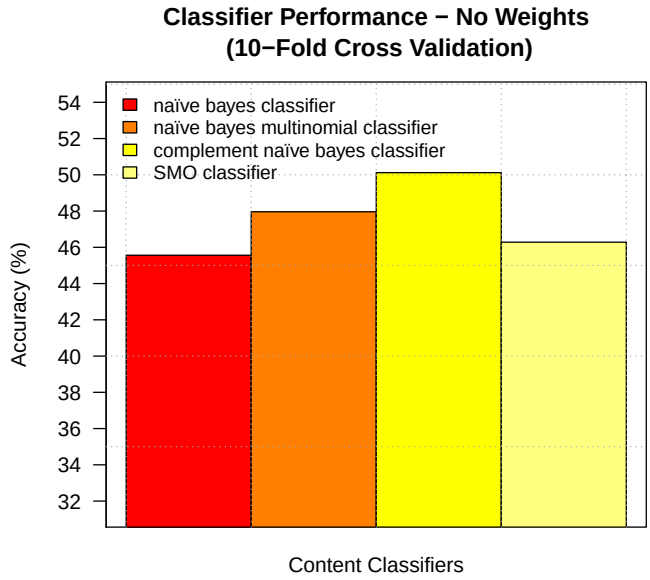


Figure 5.7: *The accuracy of the content classifiers, when trained on a normalized, stemmed, word-tokenized & binary-vectorized dataset, using different algorithms.*

In an effort to enhance the classifiers' accuracy, the researchers returned to the preprocessing phase. This time, in the tokenisation step, the number of words to be kept from each class was increased to 2000. A stop-words removal step was also added. For vectorisation, the tf-idf weighting scheme was used instead of the previous binary scheme. Furthermore, since the resulting vocabulary is large (863 terms), another training set was created using the same aforementioned preprocessing steps but with an additional dimensionality reduction step in which all terms with *information gain* less than zero was removed.

After re-training the classifiers on the new datasets, and measuring their accuracy, an average of 11.5% performance improvement was noted, where the accuracy increased to 52%-64%, and the highest classifier is still the one trained using *complement Naïve Bayes*, this time achieving accuracy slightly over 64%. However, there was no tangible improvement in the performance when using dimensionality reduction. This is shown in Figure 5.8 below, which reflects the content classifiers' accuracy after changing the preprocessing phase.

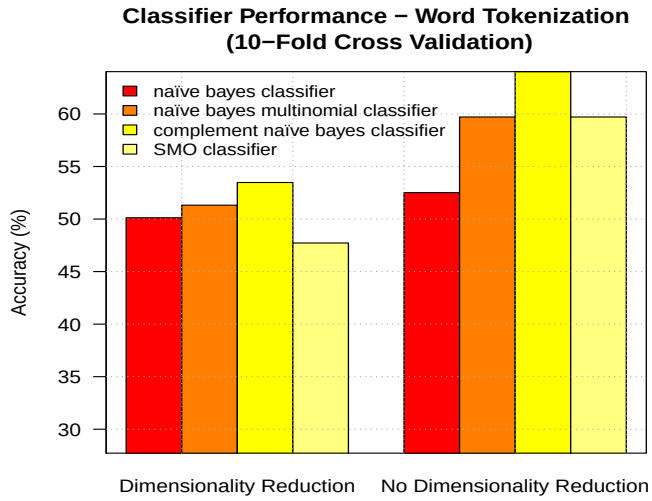


Figure 5.8: *The accuracy of the content classifiers, after increasing the vocabulary size,removing stop words, and using tf-idf weighting scheme, with/without dimensionality reduction.*

In order to further enhance the classifiers accuracy, the preprocessing phase was returned to. This time, the same preprocessing steps that were used previously were used again, except for tokenisation, where 3-gram tokenisation was used instead of word tokenisation. The cross-validation results showed a small accuracy improvement; while highest accuracy is still achieved by the classifier trained using *complement Naïve Bayes*, which was around 65.2%. Figure 5.9 below shows the content classifiers' accuracy after the final changes to the preprocessing phase.

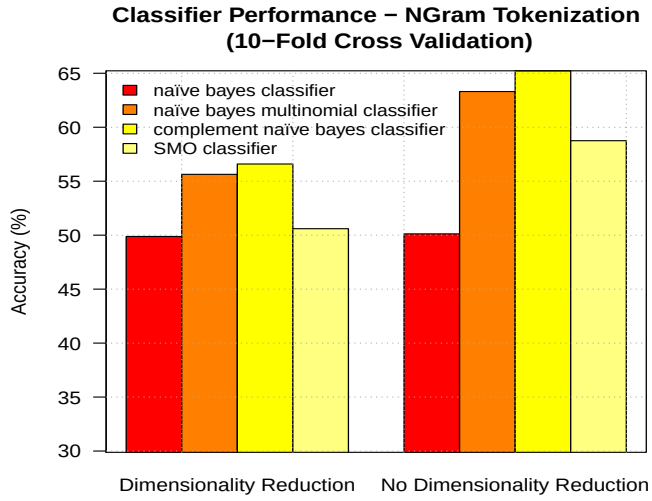


Figure 5.9: *The accuracy of the content classifiers, after increasing the vocabulary size,removing stop words, using tf-idf weighting scheme, and 3-gram tokenization, with/without dimensionality reduction.*

The experimental results showed that the highest accuracy was achieved by the classifier trained using *complement Naïve Bayes*; on a dataset preprocessed using a combination of normalisation, stemming, 3-gram tokenisation, stop-words removal, and tf-idf weighting. Therefore, this classifier was selected to be the UCP's main content classifier.

Next, in order to further evaluate the performance of the selected classifier, the chosen classifier (i.e. the *complement Naïve Bayes*) was validated on the 'untouched' testing data set Te. The results showed that the classifier's performance was relatively stable, achieving an accuracy of around 64.2%. This can be viewed in Figure 5.10 below, which shows the accuracy of the selected UCP content classifier as tested on the test dataset Te.

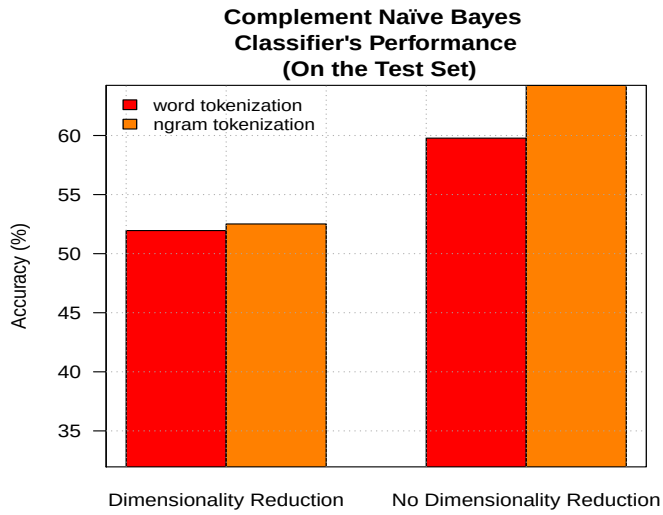


Figure 5.10: *Accuracy of selected UCP content classifier when evaluated on Te.*

In the next chapter, the work is concluded with a discussion of the experimental results, and ideas for future work are presented.

Chapter 6

Conclusion

6.1 Introduction

The goal of this research was to help SMPs users (especially inexperienced ones) configure the privacy policies of their profile attributes, as well as their generated content, while requiring minimum input from the users.

In order to meet these goals, a framework was proposed for a privacy policy recommender system that assists SMPs' users by providing them with personalised suggestions as to how they should configure the privacy policies of both of their profile attributes and contents. The recommender system was designed to be deployed on, and maintained by, the SMP provider and it consists of two independent components that work in parallel to protect users' privacy.

The first component is the profile attributes protector (PAP), which is responsible for suggesting suitable privacy policies for target users' profile attributes. The PAP follows a demographic-based approach to recommender systems, whereby it relies on the privacy policies that existing (presumably more experienced) users have specified for their profile attributes to suggest to new (presumably naïve) target users how to configure their profile attribute privacy policies. Specifically, the PAP uses the existing SMP's users' demographic information

and privacy policies to train a series of decision tree classifiers, then the PAP uses these decision tree classifiers to suggest suitable privacy policies for our target users' profile attributes.

The second component is the user content protector (UCP), which is responsible for suggesting suitable privacy policies for content generated by our target users. The PAP follows a content-based approach to recommender systems, whereby it 'learns' from the privacy policies that the target user has specified for content he/she shared in the past (i.e. privacy policy history), and then uses this acquired 'knowledge' to suggest suitable privacy policies for content that this particular target user might share in the future. Specifically, the UCP uses the target user's privacy policy history to train a Naïve Bayes classifier, and then uses this classifier to suggest suitable privacy policies for other content the target user might share in the future.

For the purpose of assessing the feasibility of the proposed privacy policy recommender system framework, the researchers experimented with implementing a basic prototype of the privacy policy recommender system, focusing primarily on the classifiers that predict users' privacy policies.

In order to implement the PAP component of the recommender system's prototype, the researchers used both simulated and real datasets for the task of training the PAP profile attribute classifiers. The results showed that when simulated datasets are used, the PAP classifiers are able to suggest privacy policies for profile attributes with a high accuracy, ranging between 60-80%. Furthermore, this accuracy can be improved by up to 25% if a combination of demographics and privacy policies are used as features. When the real dataset was used, however, the accuracy was slightly lower; ranging between 60-70%, but this is expected given the structural differences between the simulated and the real dataset. Interestingly, however, there was an average of 17% improvement

in the accuracy when a combination of demographics and privacy policies were used as features. This is indicative of the fact that the classifiers' accuracy (and thus the PAP's accuracy) rate can be significantly increased by requiring the user to provide some privacy policies for a few attributes.

In order to implement the UCP component of the recommender system's prototype; the researchers collected a dataset of posts (i.e. user-generated content) from the author's Facebook account. They then experimented with applying different combinations of preprocessing steps (e.g. word vs. n-gram tokenisation, binary vs. tf-idf weighting, with/without dimensionality reduction) on recently harvested data, to transform it from a corpus of text, to a dataset of labelled feature vectors. Next, several classification algorithms were experimentally applied to the preprocessed datasets in order to train the user-generated content classifier. The results showed that the best performing UCP content classifier was able to predict suitable privacy policies for user-generated content with an accuracy reaching about 63%.

In general, the experimental results of the privacy policy recommender system's prototype implementation were promising, as they showed that such a recommender is indeed feasible and that it (i.e. the recommender) was able to predict (and hence suggest) suitable privacy policies for both profile attributes and user-generated content with high accuracy, while requiring only minimum input from the SMP's users.

6.2 Future Work

Experimental results for implementing the recommender system's prototype were promising. However, the fact that (synthetic and incomplete) data was used for implementing the PAP, in addition to data from only one user (i.e. the

author) for implementing the UCP, might not truly reflect the reality. Therefore, a further investigation of how the recommender system will perform on different datasets is needed. For instance, implementing the PAP component using complete and real SMPs datasets, and implementing the UCP component using different datasets from different users.

Furthermore, since in this work the scope of the recommender system (the UCP component in particular) was limited to suggesting privacy policies for text-based user-generated content only; future work should involve investigating how the current approach would generalise to other types of user-generated content, like photos and videos.

Additionally, in order to train (i.e. build) the UCP content classifier, a 'batch learning' approach to machine learning was followed, where a complete training dataset existed prior to building the classifier. However, in reality target users generate content sequentially, which means there will be a period of 'idleness' where the recommender waits for the target user to generate enough training data in order for it (i.e. the recommender) to build that target user's content classifier. Therefore, an 'online learning' approach is suggested for future work when building the UCP content classifier, whereby the classifier is incrementally updated with the arrival of new training data. This way there will be no period of 'idleness'. Furthermore, if there is a clear feedback loop, the target user could be more involved in training his/her content classifier.

In general, the recommender system's classifiers managed to predict privacy policies with high accuracy. However, the performance of some classifiers still needs improvement. Therefore, future work should also involve working on improving the accuracy of the recommender system's classifiers.

Lastly, in order to evaluate the performance of the recommender system's prototype, the researchers relied on methods like cross validation to measure how

well the recommender system's classifiers can predict users' privacy policies. However, since the proposed recommender system is expected to be used by real users, future work should involve extending the evaluation mechanism by getting real users to evaluate the recommender system, instead of solely relying on methods like cross validation.

Appendix A

The SMP Simulation Model

```
;####################################
;####################################
;####  A SIMULATIION MODEL FOR AN ####
;####    SOCIAL MEDIA PLATFORM    ####
;####################################
;####################################

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;; CREATING THE USERS BREED ;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
breed[users user]

;;;;; DEFINING USERS' (I.E. AGENTS') CHARACTERISTICS ;;;;;
users-own[
  ;;;;; USERS' PROFILE ATTRIBUTES ;;;;;
  age
  gender
  university
  country
  current-city
  my-interest
  my-language
  my-relationship-status

  ;;;;; USERS' INTERNAL PROPERTIES ;;;;;
  privacy-concern  ;; indicating the users level of privacy concern.
  my-friendship-threshold   ;; indicating how "friendly" the user is.
  my-maximum-degree  ;; the maximum no. friends the user is able to maintain.
  policies-configured?

  ;;;;; USERS' PRIVACY POLICIES ;;;;;
  age-PrivacyPolicy
  gender-PrivacyPolicy
  university-PrivacyPolicy
  country-PrivacyPolicy
  current-city-PrivacyPolicy
  interests-PrivacyPolicy
  language-PrivacyPolicy
  relationship-status-PrivacyPolicy
]

;;;;; CREATING GLOBAL VARIABLES ;;;;;
globals [genders universities countries cities interests languages
  relationship-status privacy-policies attributes-weights is-random-data ]
```

```
;;;;; INITIALIZING GLOBAL VARIABLES (FUNCTION) ;;;;;
to intitialize-globals
  set genders ["male" "female"]
  set universities ["UCT" "UP" "UWC" "Wits" "Rhodes"]
  set countries ["south-africa" "sudan" "nigeria" "france" "canada"
    "Swaziland" "kenya" "cuba" "ghana"]
  set cities ["khartoum" "cape-town" "Johannesburg" "Durban"
    "Pretoria" "Mhluzi" "Port-Elizabeth"]
  set interests ["music" "sports" "reading" "dance" "programming"
    "traveling" "theater" "cinema"]
  set languages ["arabic" "english" "french"
    "xhosa" "zulu" "afrikaans"]
  set relationship-status ["single" "in-a-relationship" "married" "engaged"]
  set privacy-policies ["public" "friends-of-friends" "friends" "only-me"]
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;; PREPARING THE SIMULATION  ;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to setup
  ca ;; clearing all variables
  reset-ticks
  intitialize-globals ;; initializing global variables
  set is-random-data false

  ;;;;; NOW WE ACTUALLY CREATE THE USERS (I.E. AGENTS) ;;;;;
  create-users num-users [ ;; num-users is a simulation parameter.
  setxy random-xcor random-ycor
  set shape "circle"
  set color red

  ;;;;; ASSIGNING VALUES TO PROFILE ATTRIBUTES ;;;;;
  set age ceiling abs random-normal 30 5 ;;the age attribute is...
  ;;... distributed gaussian with mean 30 and variance 5.
  set gender item (random 2) genders ;; the attribute gender ...
  ;;...is selected at random (i.e. uniformly distributed).
  set university item (random 5) universities ;; the attribute university...
  ;;...is selected at random (i.e. uniformly distributed).
  set country item (random 9) countries ;; the attributes country ...
  ;;...is selected at random (i.e. uniformly distributed).
  set current-city item (random 7) cities ;; the attribute city ...
  ;;...is selected at random (i.e. uniformly distributed).
  set my-interest item (random 8) interests ;; the attribute interests...
  ;;...is selected at random (i.e. uniformly distributed).
  set my-language item (random 6) languages ;; random language
  set my-relationship-status item (random 4) relationship-status ;;attribute...
```

```
  ;;... relationship-status is selected at random (i.e. uniformly distributed).

  ;;;;; ASSIGNING VALUES TO USERS' INTERNAL PROPERTIES ;;;;;
  set privacy-concern abs random-normal .5 .15 ;; the privacy-concern livel...
  ;; is distributed gaussian with mean .5 and variance ,15.
  set my-friendship-threshold abs random-normal .5 .15 ;; the friendship...
  ;; threshold is distributed gaussian with mean .5 and variance ,15.
  set my-maximum-degree ceiling abs random-normal 0 10 ;; the maximum degree...
  ;; is distributed gaussian with mean 0 and variance 15.
  set policies-configured? false

  ;;;;; ASSIGNING THE DEFAULT PRIVACY POLICIES (EVERYTHING IS PUBILIC) ;;;;;
  set age-PrivacyPolicy "public"
  set gender-PrivacyPolicy "public"
  set university-PrivacyPolicy "public"
  set country-PrivacyPolicy "public"
  set current-city-PrivacyPolicy "public"
  set interests-PrivacyPolicy "public"
  set language-PrivacyPolicy "public"
  set relationship-status-PrivacyPolicy "public"
  set is-random-data false
  ]
end

;;;;; COUNTING FRIEND-IN-COMMON (FUNCTION) ;;;;;
to-report friend-in-common [source dest]
  ;; initially no friends in common
  let result 0
  let flag false

  ;; count the friends in common
  ask [link-neighbors] of source
  [
    set flag true
    if (member? dest link-neighbors) [set result result + 1]
  ]

  ;; return the percentage of the friend in common.
  ifelse flag [report result / count [link-neighbors] of source][report 0]
end

;;;;; SOCIALIZE: FORMING THE FRIENDSHIP-GRAPH (FUNCTION) ;;;;;
to socialize
 tick
 set-current-plot "No. Users who Configured Their Settings"
 clear-plot
```

```
 layout-spring users links .5 10 5 ;; making the layout of the graph.

 ask users [
   ;; check if this user have reached his internal maximum no. friends.
   if( (count link-neighbors) <= my-maximum-degree)[
    let me self
    ;; the set of all possible friends.
    let possible-friends other turtles with [not member? self
      [link-neighbors] of me]

    ;; pick one possible friend at random.
    let new-friend one-of possible-friends
    ;; calculate the mutual friend percentage.
    let mutual-friends-percentage friend-in-common me new-friend
    ;; calculate the similarity percentage.
    let attribute-similarity similarity me new-friend

    ;; calculate the friendship score.
    let frienship-score ( (.7 * attribute-similarity)
      + (.3 * mutual-friends-percentage) )

    ;; finally decide whether to form a friendship or not.
      if ( frienship-score >= [my-friendship-threshold] of me) [
        create-link-with new-friend
     ]
   ]
 ]
 ;; if all users have made enough friendships, stop socializing.
 if(not any? users with [ (count link-neighbors) <= my-maximum-degree])
 [stop]
end


;;;;; SIMILARITY BETWEEN USERS (FUNCTION) ;;;;;
to-report similarity [me otheruser]
  ;; initializing the similarity result
  let sim-result 0

  if ([age] of me = [age] of otheruser )
  [set sim-result  sim-result + 1]
  if ([gender] of me = [gender] of otheruser )
  [set sim-result  sim-result + 1]
  if ([university] of me = [university] of otheruser )
  [set sim-result  sim-result + 1]
  if ([country] of me = [country] of otheruser )
  [set sim-result  sim-result + 1]
  if ([current-city] of me = [current-city] of otheruser )
```

```
  [set sim-result  sim-result + 1]
  if ([my-interest] of me = [my-interest] of otheruser )
  [set sim-result  sim-result + 1]
  if ([my-language] of me = [my-language] of otheruser )
  [set sim-result  sim-result + 1]
  if ([my-relationship-status] of me = [my-relationship-status] of
    otheruser )
  [set sim-result  sim-result + 1]
  if ( ([privacy-concern] of me - [privacy-concern] of otheruser) < .05 )
  [set sim-result  sim-result + 1]

  ;; calculate the similarity percentage
  set sim-result sim-result / 9

  report sim-result
end


;;;;; SIMULATE PRIVACY POLICIES (FUNCTION) ;;;;;
to simulate-privacy-policies
ifelse (count users with [ not policies-configured? ] >= 10) [
  tick

  ;; ask ten random user to concider configure their privacy policies.
  ask n-of (random 10) users with [ not policies-configured? ]  [

  set color yellow

  ;; calculating user's motivation to configure his/her privacy policies.
  let friends-who-configuered-policies 0
  if any? link-neighbors [ set friends-who-configuered-policies (count
      link-neighbors with [policies-configured?] / count link-neighbors ) ]
  let motivation ( alpha * privacy-concern +  beta *
    friends-who-configuered-policies )

  ;;if the user is motivated enough, configure the privacy policies.
  ifelse motivation >= 0.30 [ configure-privacy-policies self][ set color red ]

  ]
  ][stop]
end

;;;;; CONFIGURING PRIVACY POLICIES (FUNCTION) ;;;;;
to configure-privacy-policies [ selected-users ]
  ask selected-users [ ;; alpha, beta, and the weights are simulation parameters.
    ;; configuring the privacy settins for individual profile attributes.
    set age-PrivacyPolicy get-privacy-policy (
```

```
      (alpha * privacy-concern) + (beta * age-weight) )
    set gender-PrivacyPolicy get-privacy-policy (
      (alpha * privacy-concern) + (beta * gender-weight) )
    set university-PrivacyPolicy get-privacy-policy (
      (alpha * privacy-concern) + (beta * university-weight) )
    set country-PrivacyPolicy get-privacy-policy (
      (alpha * privacy-concern) + (beta * country-weight) )
    set current-city-PrivacyPolicy get-privacy-policy (
      (alpha * privacy-concern) + (beta * current-city-weight) )
    set interests-PrivacyPolicy get-privacy-policy (
      (alpha * privacy-concern) + (beta * interest-weight) )
    set language-PrivacyPolicy get-privacy-policy (
      (alpha * privacy-concern) + (beta * language-weight) )
    set relationship-status-PrivacyPolicy get-privacy-policy (
      (alpha * privacy-concern) + (beta * relationship-weight) )

    ;; making these users visually traceable.
    set color white
    set policies-configured? true
    ]
end

;;;;; GET POLICIES VALUE (FUNCTION) ;;;;;
to-report get-privacy-policy [ input ]
  if ( input >= 0.00 and input <= 0.25) [report item 0 privacy-policies]
  if ( input >  0.25 and input <= 0.5 ) [report item 1 privacy-policies]
  if ( input >  0.50 and input <= 0.75) [report item 2 privacy-policies]
  if ( input >  0.75                  ) [report item 3 privacy-policies]
end

;;;;;  RESET PRIVACY POLICIES (FUNCTION)  ;;;;;
to reset
ask users [
    set policies-configured? false
    set age-PrivacyPolicy "public"
    set gender-PrivacyPolicy "public"
    set university-PrivacyPolicy "public"
    set country-PrivacyPolicy "public"
    set current-city-PrivacyPolicy "public"
    set interests-PrivacyPolicy "public"
    set language-PrivacyPolicy "public"
    set relationship-status-PrivacyPolicy "public"
    set color red
  ]
  set-current-plot "No. Users who Configured Their Settings"
 clear-plot
end
```

```
;;;;; GENERATING RANDOM PRIVACY POLICIES (FUNCTION) ;;;;;
to random-policies
  setup
  set is-random-data true
  ask users [
    set age-PrivacyPolicy item (random 4) privacy-policies
    set gender-PrivacyPolicy item (random 4) privacy-policies
    set university-PrivacyPolicy item (random 4) privacy-policies
    set country-PrivacyPolicy item (random 4) privacy-policies
    set current-city-PrivacyPolicy item (random 4) privacy-policies
    set interests-PrivacyPolicy item (random 4) privacy-policies
    set language-PrivacyPolicy item (random 4) privacy-policies
    set relationship-status-PrivacyPolicy item (random 4) privacy-policies
    set policies-configured? true
    set color white
    ]
end


;;;;;;;;;;;;;;;;;;;;;;
;;;;; STATISTICS ;;;;;
;;;;;;;;;;;;;;;;;;;;;;

to-report average-degree
  report round ( mean [(count link-neighbors)] of users )
end

to-report max-degree
   report max [(count link-neighbors)] of users
end

to-report min-degree
   report min [(count link-neighbors)]  of users
end

to-report no-of-friendships
  report  sum [(count link-neighbors)] of users / 2
end
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;; EXPORTING SIMULATION DATA FOR ANALYSIS ;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to export-data
  let time date-and-time
  set time (word substring date-and-time 0 8 "@" substring
    date-and-time 16 27)
  set time replace-item 2 time "-"
  set time replace-item 5 time "-"


  let extention ""
  let rand ""
  let graph-file-name (word "Exported Files\\graph-[" num-users "]-"
    time ".gexf")

  if (output-file-type = "weka-arff") [
    set extention ".arff"
    ]
  if (output-file-type = "csv" ) [
    set extention ".csv"
    ]
  if (output-file-type = "general") [
    set extention ".txt"
    ]

  if(is-random-data) [set rand "random-"]
  let dataset-file-name (word "Exported Files\\" rand "dataset-["
    num-users "]-" time)

  ;;;;; GEXF FILE  ;;;;;
  file-open ( graph-file-name )
  file-print (word "<?xml version="  "'1.0'"  "encoding='UTF-8'" "?>")
  file-print (word "<gexf"  " xmlns='https://people.cs.uct.ac.za/~aabuelgasim'"
    "version='1.2'" " >")
  file-print (word "<meta lastmodifieddate="  "'timer'"  " >" )
  file-print (word "<creator>Ammar M. Abuelgasim</creator>")
  file-print (word "<description> Simulated Social "
    "Network Dataset </description>")
  file-print (word "</meta>")
  file-print (word "<graph mode=" "'static'" "defaultedgetype='undirected'" ">")
  file-print  "<nodes>"

  ask users [
    file-print (word "<node id='" who "'label='" who "' />")
    ]
```

```
file-print "</nodes>"
file-print "<edges>"

let link-counter 0
ask links [
  file-print (word "<edge id='" end1  link-counter "' source='"
    [who] of end1 "' target='"  [who] of end2 "'/>")
  set link-counter link-counter + 1
  ]

file-print "</edges>"
file-print "</graph>"
file-print "</gexf>"

file-close
;;;;; GEXF FILE END ;;;;;


;;;;; ARFF FILE ;;;;;
if (output-file-type = "weka-arff" or output-file-type = "all") [
file-open( (word dataset-file-name ".arff" ))
file-print (word "@relation " "Simulated-"
  (substring dataset-file-name 15 49) " \r\n" )
file-print " \r\n"

;;feature attributes
file-print (word "@attribute age real \r\n")
file-print (word "@attribute gender {male, female} \r\n")
file-print (word "@attribute university {UCT, UP, UWC, Wits, Rhodes} \r\n")
file-print (word "@attribute country {south-africa, sudan, "
  "nigeria, france, canada, Swaziland, kenya, cuba, ghana} \r\n")
file-print (word "@attribute current-city {khartoum, cape-town, "
  "Johannesburg, Durban, Pretoria, Mhluzi, Port-Elizabeth} \r\n")
file-print (word "@attribute my-interest {music, sports, reading, "
  "dance, programming, traveling, theater, cinema} \r\n")
file-print (word "@attribute my-language {arabic, english, "
  "french, xhosa, zulu, afrikaans} \r\n")
file-print (word "@attribute my-relationship-status {single, "
  "in-a-relationship, married, engaged} \r\n")
;;class attributes
file-print (word "@attribute age-PrivacyPolicy {public, "
  "friends-of-friends, friends, only-me} \r\n")
file-print (word "@attribute gender-PrivacyPolicy {public, "
  "friends-of-friends, friends, only-me} \r\n")
file-print (word "@attribute university-PrivacyPolicy {public, "
  "friends-of-friends, friends, only-me} \r\n")
file-print (word "@attribute country-PrivacyPolicy {public, "
```

```
    "friends-of-friends, friends, only-me} \r\n")
  file-print (word "@attribute current-city-PrivacyPolicy {public, "
    "friends-of-friends, friends, only-me} \r\n")
  file-print (word "@attribute interests-PrivacyPolicy {public, "
    "friends-of-friends, friends, only-me} \r\n")
  file-print (word "@attribute language-PrivacyPolicy {public, "
    "friends-of-friends, friends, only-me} \r\n")
  file-print (word "@attribute relationship-status-PrivacyPolicy "
    "{public, friends-of-friends, friends, only-me} \r\n")
  file-print " \r\n"

  file-print "@data \r\n"
ask users with [policies-configured?] [
  file-print (word  age "," gender "," university "," country  "," current-city
  "," my-interest  "," my-language ","  my-relationship-status ","
  age-PrivacyPolicy "," gender-PrivacyPolicy  "," university-PrivacyPolicy
  "," country-PrivacyPolicy  "," ","  relationship-status-PrivacyPolicy "\r\n" )
   ]
  file-close
  ]
;;;;; ARFF FILE's END ;;;;;

;;;;; CSV FILE ;;;;;
if(output-file-type = "csv" or output-file-type = "all") [
  file-open( (word dataset-file-name ".csv" ))
  file-print (word "age,gender,university,country,current-city,my-interest,"
  "my-language,my-relationship-status,age-PrivacyPolicy,gender-PrivacyPolicy,"
  "university-PrivacyPolicy,country-PrivacyPolicy,current-city-PrivacyPolicy,"
  "interests-PrivacyPolicy,language-PrivacyPolicy,"
  "relationship-status-PrivacyPolicy" "\r\n" )

 ask users with [policies-configured?] [
    file-print (word age "," gender "," university "," country  "," current-city ","
    my-interest  "," my-language ","  my-relationship-status "," age-PrivacyPolicy ","
    gender-PrivacyPolicy  "," university-PrivacyPolicy  "," country-PrivacyPolicy  ","
    current-city-PrivacyPolicy ","  interests-PrivacyPolicy ","  language-PrivacyPolicy
    "," relationship-status-PrivacyPolicy "\r\n" )
    ]
     file-close
  ]
 ;;;;; CSV FILE END ;;;;;

;;;;; TAKING A SCREENSHOT ;;;;;
Export-Interface (word dataset-file-name ".png" )
user-message (word "File Exported Successfully!")
end
```

Bibliography

[1] ACQUISTI, A., CARRARA, E., STUTZMAN, F., CALLAS, J., SCHIMMER, K., NADJM, M., GORGE, M., ELLISON, N., KING, P., GROSS, R., AND GOLDER, S. Security Issues and Recommendations for Online Social Networks. Tech. Rep. 1, European Network and Information Security Agency, 2007. Availabel at https://goo.gl/PBdelw.

[2] AGGARWAL, C. C., AND ZHAI, C. A Survey of Text Classification Algorithms. In *Mining Text Data*. Springer US, Boston, MA, 2012, pp. 163–222. Available at http://link.springer.com/10.1007/978-1-4614-3223-4_6.

[3] AKCORA, C., CARMINATI, B., AND FERRARI, E. Privacy in social networks: How risky is your social graph? In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on* (April 2012), pp. 9–19.

[4] ALSALIBI, B., AND ZAKARIA, N. CFPRS : Collaborative Filtering Privacy Recommender System for Online Social Networks. *Journal of Engineering Research and Applications 3*, 5 (2013), 1850–1858. Available at http://www.ijera.com/papers/Vol3_issue5/KT3518501858.pdf.

[5] ALTMAN, I. Privacy regulation: Culturally universal or culturally specific? *Journal of Social Issues 33*, 3 (1977), 66–84. Avialable at http://dx.doi.org/10.1111/j.1540-4560.1977.tb01883.x.

[6] ANG, C., AND ZAPHIRIS, P. Simulating social networks of online communities: Simulation as a method for sociability design. In *Human-Computer*

Interaction – INTERACT 2009, T. Gross, J. Gulliksen, P. Kotzé, L. Oestreicher, P. Palanque, R. Prates, and M. Winckler, Eds., vol. 5727 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2009, pp. 443–456. Available at http://dx.doi.org/10.1007/978-3-642-03658-3_48.

[7] BACKSTROM, L., DWORK, C., AND KLEINBERG, J. Wherefore art thou r3579x?: Anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th International Conference on World Wide Web* (New York, NY, USA, 2007), WWW '07, ACM, pp. 181–190. Available at http://doi.acm.org/10.1145/1242572.1242598.

[8] BAKSHY, E., ECKLES, D., YAN, R., AND ROSENN, I. Social influence in social advertising: Evidence from field experiments. In *Proceedings of the 13th ACM Conference on Electronic Commerce* (New York, NY, USA, 2012), EC '12, ACM, pp. 146–161. Available at http://doi.acm.org/10.1145/2229012.2229027.

[9] BECKER, J., AND CHEN, H. *Measuring privacy risk in online social networks*. PhD thesis, May 2009. Available at http://web.cs.ucdavis.edu/~hchen/paper/w2sp2009.pdf.

[10] BESMER, A., LIPFORD, H. R., SHEHAB, M., AND CHEEK, G. Social applications: Exploring a more secure framework. In *Proceedings of the 5th Symposium on Usable Privacy and Security* (New York, NY, USA, 2009), SOUPS '09, ACM, pp. 2:1–2:10. Avialable at http://doi.acm.org/10.1145/1572532.1572535.

[11] BILGE, L., STRUFE, T., BALZAROTTI, D., AND KIRDA, E. All your contacts are belong to us: Automated identity theft attacks on social networks. In *Proceedings of the 18th International Conference on World Wide Web*

(New York, NY, USA, 2009), WWW '09, ACM, pp. 551–560. Available at http://doi.acm.org/10.1145/1526709.1526784.

[12] BONNEAU, J., ANDERSON, J., ANDERSON, R., AND STAJANO, F. Eight friends are enough: Social graph approximation via public listings. In *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems* (New York, NY, USA, 2009), SNS '09, ACM, pp. 13–18. Available at http://doi.acm.org/10.1145/1578002.1578005.

[13] BOYD, D. M., AND ELLISON, N. B. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication 13*, 1 (2007), 210–230. Available at http://dx.doi.org/10.1111/j.1083-6101.2007.00393.x.

[14] CAMPAN, A., AND TRUTA, T. Data and structural k-anonymity in social networks. In *Privacy, Security, and Trust in KDD*, F. Bonchi, E. Ferrari, W. Jiang, and B. Malin, Eds., vol. 5456 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2009, pp. 33–54. Available at http://dx.doi.org/10.1007/978-3-642-01718-6_4.

[15] CARMINATI, B., FERRARI, E., HEATHERLY, R., KANTARCIOGLU, M., AND THURAISINGHAM, B. A semantic web based framework for social network access control. In *Proceedings of the 14th ACM Symposium on Access Control Models and Technologies* (New York, NY, USA, 2009), SACMAT '09, ACM, pp. 177–186. Available at http://doi.acm.org/10.1145/1542207.1542237.

[16] CARMINATI, B., FERRARI, E., HEATHERLY, R., KANTARCIOGLU, M., AND THURAISINGHAM, B. Semantic web-based social network access control. *Computers & Security 30*, 2–3 (2011), 108 – 115. Special Issue on Access Control Methods and Technologies.

[17] CARMINATI, B., FERRARI, E., AND PEREGO, A. Enforcing access control in web-based social networks. *ACM Trans. Inf. Syst. Secur. 13*, 1 (Nov. 2009), 6:1–6:38. http://doi.acm.org/10.1145/1609956.1609962.

[18] CHENG, Y., PARK, J., AND SANDHU, R. Relationship-Based Access Control for Online Social Networks: Beyond User-to-User Relationships. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing* (Sept. 2012), IEEE, pp. 646–655. Available at http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6406322.

[19] CHENG, Y., PARK, J., AND SANDHU, R. A user-to-user relationship-based access control model for online social networks, 2012. Available at http://dx.doi.org/10.1007/978-3-642-31540-4_2.

[20] CHENG, Y., PARK, J., AND SANDHU, R. Preserving user privacy from third-party applications in online social networks. 723–728. Available at http://dl.acm.org/citation.cfm?id=2487788.2488032.

[21] DANEZIS, G. Inferring privacy policies for social networking services. In *Proceedings of the 2Nd ACM Workshop on Security and Artificial Intelligence* (New York, NY, USA, 2009), AISec '09, ACM, pp. 5–10. Available at http://doi.acm.org/10.1145/1654988.1654991.

[22] DO, H. G., NG, W. K., AND MA, Z. Privacy-Preserving Social Network for an Untrusted Server. In *2013 International Conference on Cloud and Green Computing* (Sept. 2013), IEEE, pp. 472–478. Available at http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6686072.

[23] EFFENDY, S., YAP, R. H., AND HALIM, F. Revisiting link privacy in social networks. In *Proceedings of the Second ACM Conference on Data and*

Application Security and Privacy (New York, NY, USA, 2012), CODASPY '12, ACM, pp. 61–70. Available at http://doi.acm.org/10.1145/2133601.2133609.

[24] FACEBOK. Company Information, 2015. https://newsroom.fb.com/company-info/ , Last accessed on (2015-10-13).

[25] FANG, L., AND LEFEVRE, K. Privacy wizards for social networking sites. In *Proceedings of the 19th International Conference on World Wide Web* (New York, NY, USA, 2010), WWW '10, ACM, pp. 351–360. Available at http://doi.acm.org/10.1145/1772690.1772727.

[26] FELT, A., AND EVANS, D. Privacy Protection for Social Networking APIs. *Workshop on Web 2.0 Security and Privacy, Oakland, CA* (2008). Available at http://www.cs.virginia.edu/felt/privacybyproxy.pdf.

[27] FIRE, M., GOLDSCHMIDT, R., AND ELOVICI, Y. Online social networks: Threats and solutions. *Communications Surveys Tutorials, IEEE 16*, 4 (Fourthquarter 2014), 2019–2036. Available at http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6809839.

[28] GAO, H., HU, J., HUANG, T., WANG, J., AND CHEN, Y. Security issues in online social networks. *Internet Computing, IEEE 15*, 4 (July 2011), 56–63. Available at http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5740826.

[29] GHAZINOUR, K., MATWIN, S., AND SOKOLOVA, M. Monitoring and recommending privacy settings in social networks. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops* (New York, NY, USA, 2013), EDBT '13, ACM, pp. 164–168. Available at http://doi.acm.org/10.1145/2457317.2457344.

[30] GHAZINOUR, K., MATWIN, S., AND SOKOLOVA, M. YourPrivacyProtector: A Recommender System for Privacy Settings in Social Networks. *International Journal of Security 2*, 4 (2013), 11–25. Available at http://www.airccse.org/journal/ijsptm/papers/2413ijsptm02.pdf.

[31] GNIP. Social Media Monitoring, 2015. https://gnip.com/industries/social-media-monitoring/ , Last accessed on (25-06-2015).

[32] GROSS, R., AND ACQUISTI, A. Information revelation and privacy in online social networks. In *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society* (New York, NY, USA, 2005), WPES '05, ACM, pp. 71–80. Available at http://doi.acm.org/10.1145/1102199.1102214.

[33] GUO, S., AND CHEN, K. Mining privacy settings to find optimal privacy-utility tradeoffs for social network services. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)* (Sept 2012), IEEE, pp. 656–665. Available at http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6406323.

[34] HALL, M., FRANK, E., HOLMES, G., PFAHRINGER, B., REUTEMANN, P., AND WITTEN, I. H. The weka data mining software: An update. *SIGKDD Explor. Newsl. 11*, 1 (Nov. 2009), 10–18. http://www.cs.waikato.ac.nz/ml/weka/.

[35] HEIDEMANN, J., KLIER, M., AND PROBST, F. Online social networks: A survey of a global phenomenon. *Computer Networks 56*, 18 (2012), 3866–3878. Available at http://dx.doi.org/10.1016/j.comnet.2012.08.009.

[36] HIRALALL, M. Recommender systems for e-shops. Tech. Rep.

1663100, University of Amsterdam, Amsterdam, 2011. Available at http://www.few.vu.nl/en/Images/werkstuk-hiralall_tcm39-202691.pdf.

[37] JIN, L., TAKABI, H., LONG, X., AND JOSHI, J. Exploiting users' inconsistent preferences in online social networks to discover private friendship links. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society* (New York, NY, USA, 2014), WPES '14, ACM, pp. 59–68. Available at http://doi.acm.org/10.1145/2665943.2665956.

[38] JONES, S. *Automating Group-Based Privacy Control in Social Networks*. Doctor of philosophy, University of Bath, 2012.

[39] KLIMEK, P., AND THURNER, S. Triadic closure dynamics drives scaling laws in social multiplex networks. *New Journal of Physics 15*, 6 (2013), 063008. Available at http://stacks.iop.org/1367-2630/15/i=6/a=063008.

[40] KONSTAN, J. Introduction to User-User Collaborative Filtering (Online Lecture), 2013. Available at https://goo.gl/vy1rjQ , Last accessed on (15-07-2015).

[41] KOSTAKOS, V., VENKATANATHAN, J., REYNOLDS, B., SADEH, N., TOCH, E., SHAIKH, S. A., AND JONES, S. Who's your best friend?: Targeted privacy attacks in location-sharing social networks. In *Proceedings of the 13th International Conference on Ubiquitous Computing* (New York, NY, USA, 2011), UbiComp '11, ACM, pp. 177–186. Available at http://doi.acm.org/10.1145/2030112.2030138.

[42] LESKOVEC, J., ANAND, R., AND ULLMAN, J. Recommendation Systems. In *Mining of Massive Datasets*. Cambridge University Press, New York, NY, USA, 2011, ch. Chapter 9, pp. 305–339. Available at http://infolab.

stanford.edu/~ullman/mmds/booka.pdf.

[43] LIU, K., AND TERZI, E. A framework for computing the privacy scores of users in online social networks. *ACM Trans. Knowl. Discov. Data 5*, 1 (Dec. 2010), 6:1–6:30. Available at http://doi.acm.org/10.1145/1870096.1870102.

[44] LIU, Y., GUMMADI, K. P., KRISHNAMURTHY, B., AND MISLOVE, A. Analyzing facebook privacy settings: User expectations vs. reality. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference* (New York, NY, USA, 2011), IMC '11, ACM, pp. 61–70. Available at http://doi.acm.org/10.1145/2068816.2068823.

[45] MACAL, C., AND NORTH, M. Tutorial on agent-based modeling and simulation. In *Proceedings of the Winter Simulation Conference, 2005.* (2005), IEEE, pp. 2–15. Available at http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1574234.

[46] MACAL, C. M., AND NORTH, M. J. Introductory tutorial: Agent-based modeling and simulation. In *Proceedings of the 2011 Winter Simulation Conference (WSC)* (Dec. 2011), IEEE, pp. 1451–1464. Available at http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6147864.

[47] MADEJSKI, M., JOHNSON, M., AND BELLOVIN, S. A study of privacy settings errors in an online social network. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on* (March 2012), IEEE, pp. 340–345. Available at http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6197507.

[48] MASOUMZADEH, A., AND JOSHI, J. Osnac: An ontology-based access

control model for social networking systems. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on* (Aug 2010), pp. 751–759. Available at http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5591484&tag=1.

[49] MITCHELL, T. M. Decision Tree Learning. In *Machine Learning*, C. L. B. Tucker, Ed., 1 ed. McGraw-Hill, Inc., New York, NY, USA, 1997, ch. 3, pp. 52–80.

[50] PESCE, J. A. P., CASAS, D. L., RAUBER, G., AND ALMEIDA, V. Privacy attacks in social media using photo tagging networks: A case study with facebook. In *Proceedings of the 1st Workshop on Privacy and Security in Online Social Media* (New York, NY, USA, 2012), PSOSM '12, ACM, pp. 4:1–4:8. Available to http://doi.acm.org/10.1145/2185354.2185358.

[51] RICCI, F., ROKACH, L., SHAPIRA, B., AND KANTOR, P. B. *Recommender Systems Handbook*, 1st ed. Springer-Verlag New York, Inc., New York, NY, USA, 2010. Available at http://link.springer.com/10.1007/978-0-387-85820-3.

[52] SÁNCHEZ, D., AND VIEJO, A. Privacy Risk Assessment of Textual Publications in Social Networks. pp. 236–241.

[53] SEBASTIANI, F. Machine learning in automated text categorization. *ACM Comput. Surv. 34*, 1 (Mar. 2002), 1–47. Available at http://doi.acm.org/10.1145/505282.505283.

[54] SHAN, Z., CAO, H., LV, J., YAN, C., AND LIU, A. Enhancing and identifying cloning attacks in online social networks. In *Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication* (New York, NY, USA, 2013), ICUIMC '13, ACM, pp. 59:1–59:6. Available at http://doi.acm.org/10.1145/2448556.2448615.

[55] SHEHAB, M., CHEEK, G., TOUATI, H., SQUICCIARINI, A. C., AND CHENG, P.-C. Learning based access control in online social networks. In *Proceedings of the 19th International Conference on World Wide Web* (New York, NY, USA, 2010), WWW '10, ACM, pp. 1179–1180. Available at http://doi.acm.org/10.1145/1772690.1772863.

[56] SINHA, A., LI, Y., AND BAUER, L. What you want is not what you get: Predicting Sharing Policies for Text-based Content on Facebook. In *Proceedings of the 2013 ACM workshop on Artificial intelligence and security - AISec '13* (New York, New York, USA, 2013), ACM Press, pp. 13–24. Available at http://dl.acm.org/citation.cfm?doid=2517312.2517317.

[57] SQUICCIARINI, A. C., SUNDARESWARAN, S., LIN, D., AND WEDE, J. A3p: Adaptive policy prediction for shared images over popular content sharing sites. In *Proceedings of the 22Nd ACM Conference on Hypertext and Hypermedia* (New York, NY, USA, 2011), HT '11, ACM, pp. 261–270. Available at http://doi.acm.org/10.1145/1995966.1996000.

[58] SRIVASTAVA, A., AND GEETHAKUMARI, G. Measuring privacy leaks in online social networks. In *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on* (Aug 2013), pp. 2095–2100.

[59] STEINFIELD, C., ELLISON, N., AND LAMPE, C. Social capital, self-esteem, and use of online social network sites: A longitudinal analysis. *Journal of Applied Developmental Psychology 29*, 6 (Nov. 2008), 434–445. Available at http://linkinghub.elsevier.com/retrieve/pii/S0193397308000701.

[60] STUTZMAN, F., GROSS, R., AND ACQUISTI, A. Silent Listeners: The Evolution of Privacy and Disclosure on Facebook. *Journal of Privacy and Confidentiality 4*, 2 (2013), 7–41. Available at http://repository.cmu.edu/

jpc/vol4/iss2/2.

[61] TALUKDER, N., OUZZANI, M., ELMAGARMID, A., ELMELEEGY, H., AND YAKOUT, M. Privometer: Privacy protection in social networks. In *Data Engineering Workshops (ICDEW), 2010 IEEE 26th International Conference on* (March 2010), pp. 266–269.

[62] TOCH, E., SADEH, N. M., AND HONG, J. Generating default privacy policies for online social networks. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2010), CHI EA '10, ACM, pp. 4243–4248. Available at http://doi.acm.org/10.1145/1753846.1754133.

[63] TWITTER. About The Company, 2015. https://about.twitter.com/company , Last accessed on (2015-10-13).

[64] VAN DEN BERG, B., AND LEENES, R. Audience segregation in social network sites. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on* (Aug 2010), IEEE, pp. 1111–1116. Available at http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5590533.

[65] WANG, N., XU, H., AND GROSSKLAGS, J. Third-party apps on facebook: Privacy and the illusion of control. 4:1–4:10. Avialable at http://doi.acm.org/10.1145/2076444.2076448.

[66] WANG, Y., NEPALI, R., AND NIKOLAI, J. Social network privacy measurement and simulation. In *Computing, Networking and Communications (ICNC), 2014 International Conference on* (Feb 2014), pp. 802–806.

[67] WEI, Q., AND LU, Y. Preservation of privacy in publishing social network data. *Proceedings of the International Symposium on Electronic Commerce and Security, ISECS 2008* (2008), 421–425.

[68] WELLMAN, B. An Electronic Group is Virtually a Social Network. In *Culture of the Internet*, S. Kiesler, Ed., 1st ed. Psychology Press, 1997, ch. 9, pp. 179–205. Available at https://goo.gl/f6Glzr.

[69] WILENSKY, U. NetLogo ABMS Environment, The Center for Connected Learning and Computer-Based Modeling (CCL), 2015. https://ccl.northwestern.edu/netlogo/index.shtml , Last accessed on (05-08-2015).

[70] YANG, Y., LUTES, J., LI, F., LUO, B., AND LIU, P. Stalking online: On user privacy in social networks. In *Proceedings of the Second ACM Conference on Data and Application Security and Privacy* (New York, NY, USA, 2012), CODASPY '12, ACM, pp. 37–48. Available at http://doi.acm.org/10.1145/2133601.2133607.

[71] YUAN, M., CHEN, L., AND YU, P. S. Personalized privacy protection in social networks. *Proc. VLDB Endow. 4*, 2 (Nov. 2010), 141–150. Available at http://dx.doi.org/10.14778/1921071.1921080.

[72] ZHELEVA, E., AND GETOOR, L. To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of the 18th International Conference on World Wide Web* (New York, NY, USA, 2009), WWW '09, ACM, pp. 531–540. Available at http://doi.acm.org/10.1145/1526709.1526781.

[73] ZIMMER, M. "but the data is already public": On the ethics of research in facebook. *Ethics and Inf. Technol. 12*, 4 (Dec. 2010), 313–325. Available at http://dx.doi.org/10.1007/s10676-010-9227-5.